The Information Bottleneck: Theory and Applications

Thesis submitted for the degree "Doctor of Philosophy"

Noam Slonim

Submitted to the Senate of the Hebrew University in the year 2002

This work was carried out under the supervision of

Prof. Naftali Tishby

This thesis is dedicated to the memory of my father

Acknowledgments

First, I would like to express my deep gratitude to my advisor, Naftali Tishby. Tali has been, and still is, a constant source of inspiration for me. His wisdom, patience and inexhaustible knowledge have guided me safely through the last few years. This thesis is based on numerous fruitful conversations we have had, which in particular resulted in eleven papers we collaborated on. Moreover, many sections of this thesis have not been published elsewhere and are a direct result of our joint long discussions. His contribution to this thesis, and to me, cannot be overemphasized.

Nir Friedman has been like a second advisor to me. I have had many enjoyable and fruitful discussions with him which have led to three published papers, and hopefully many more to come. He showed me how to combine Bayesian Networks with the Information Bottleneck and made the first important step toward establishing the multivariate IB framework, which constitutes Part III of this thesis.

Yoram Singer guided me through the weird world of Information Retrieval. Moreover, whenever I needed advice, or some help, he was there to provide it.

Yair Weiss made the first rigorous step to show the relationship between Maximum Likelihood and the Information Bottleneck. Amazingly, he even succeeded in explaining it to me, and perhaps even more surprisingly, convinced me that it is true. Our collaboration was both pleasant and educational, and resulted in a joint paper which is presented in Appendix A.

Bill Bialek hosted me at NEC research institute at Princeton, for ten exciting days in May 1998, which both Tali Tishby and he used to introduce me to the Information Bottleneck idea. Thanks to him, this was the turning point of my studies. Until then, I was under the impression that I could earn my Ph.D. by writing sitcom scripts. I'm grateful to Bill for this visit which eventually determined the course of my thesis.

I would like to thank the Ph.D. students of the Machine Learning group at the Hebrew University for their friendship, support and significant help during this thesis. They include: Yoseph Barash, Gill Bejerano (who I suspect still thinks I'm using his desk), Gal Chechick, Amir Globerson, Gal Elidan (who patiently answered all my questions about Bayesian inference), Tommy Kaplan, Beata Klebanov, Ori Mosenzon (who I hope will eventually find one Dojo to train in), Iftach Nachman, Amir Navot, Matan Ninio (who configured my computer every time I asked him, and in some cases even when I did not ask him), Dana Pe'er, Eilon Portugaly, Yevgeny Seldin and Shai Shalev-Shwartz. Special thanks are due to Koby Crammer, my great roommate in Ross 61, and to Ran Gilad-Bachrach for bailing me out scientifically every time I needed it.

I would also like to express my gratitude to former members of the Machine Learning group: Itay Gat, Daniel Gil, Ran El-Yaniv, Yuval Feinstein, Shai Fine, Elad Schneidman, Adi Schreibman, Lidror Troyansky, and Golan Yona. Each and every one of them has supported me in different ways during the last few years. Special thanks are due to Leonid Kontorovich for his help in many different occasions.

I had fascinating exchanges about galaxies with Ofer Lahav from the Institute of Astronomy at the University of Cambridge, and with Rachel Somerville, currently at the University of Michigan. These talks resulted in a joint paper and many other unpublished material that hopefully will find its way to publication some day. I'm especially grateful to Ofer for hosting me in Cambridge during our joint work, and in the workshop on spectra and redshift surveys on the island of Porquerolles off the French Riviera (Astrophysicists certainly know how to pick conference locations).

I very much enjoyed collaborating with Yonatan Bilu and Michal Linial. However, I was not able to find any relations between our joint work and my thesis, so I'll simply cite it here [12].

Idan Segev, the Director of ICNC, and Eilon Vaadia, the Head of the ICNC Ph.D. program, were both helpful and willing any time I needed them.

Aliza Shadmi, the secretary of ICNC, has been a constant source of invaluable help while handling the necessary bureaucracies during my Ph.D. studies. Ziva Rehani, Silvi Belisha, and Regina Krizhanovsky from the administrative staff at the School of Computer Science and Engineering, have always been kind and assisting in any way they could.

Esther Singer made a great effort to correct my English in every sentence in this thesis, except for this one.

The remaining errors are mine, not hers, where I shamelessly used my worn “jargon” excuse to leave them in.

I would like to thank the Eshkol Fellowship of the Israeli Ministry of Science for its generous support during three years of my studies.

Last, and most important, my deepest gratitude are to my family:

- My mother, Nitza, for her support from the moment I was born.
- My brothers, Yochi, Yoram, and Barak, for being there for me.
- My son, Yahel, for making me laugh.
- And my beloved wife, Dana, for her love.

Preface

Two papers mark the starting point of this thesis. The first was published in 1993 by Pereira, Tishby and Lee [60] under the title "Distributional clustering of English words". In this paper the authors suggested a method for clustering words according to their distributions in particular syntactic contexts. More specifically, Pereira *et al.* represented different nouns as conditional probability distributions over verbs. The probability values were estimated to be proportional to the number of times a specific verb occurred with the specific noun in the same sentence. Pereira *at al.* further suggested measuring the similarity between different nouns through the KL divergence between the corresponding conditional verb distributions. They described a hierarchical ("soft") clustering procedure, motivated by a deterministic-annealing scheme, and provided detailed experimental results. In particular, these results demonstrated how semantically related nouns tend to be clustered together, and disambiguated nouns are naturally assigned to several clusters, corresponding to their different possible senses.

Six years later, in 1999, another paper was published by Tishby, Pereira and Bialek [82], that was entitled "The information bottleneck method". In this paper, Tishby *et al.* showed that the 1993 work was in fact a special case of a general variational principle that constitutes a new information-theoretic approach to data analysis. Given some joint distribution, $p(x, y)$, the basic idea was to search for a compact representation of X that preserves the maximum information about Y. Thus, the information that X contains about Y is squeezed through a compact "bottleneck", formed by a limited set of new representatives, or clusters. In this formulation, X and Y may correspond to any type of co-occurrence data, where analyzing co-occurrences of nouns and verbs is just one possible application of this potentially rich framework. Moreover, Tishby *et al.* suggested that their approach can be considered analogous to rate distortion theory, with an important distinction: the distortion measure does not need to be defined in advance, but rather naturally emerges from the joint statistics. They characterized the form of the optimal solution to this variational principle and showed that the deterministic-annealing approach, suggested six years earlier, can be used to construct solutions in practice.

This thesis first reviews in detail the Information Bottleneck (IB) approach and its relations to rate distortion theory. We provide precise definitions of some of the ideas that were briefly mentioned in [82]. We further suggest new algorithmic approaches to construct solutions to the IB problem and provide empirical results that demonstrate the method's usefulness in a variety of applications. Some of these applications were first presented in [74, 75, 76, 77, 78, 83] and are based on joint work with Naftali Tishby, and additionally with Nir Friedman, Ofer Lahav and Rachel Somerville. Inspired by these works, additional applications have been suggested by other authors. Several examples are presented in [38, 41, 56, 67, 68, 87].

The second half of this thesis is devoted to a theoretic extension of the IB framework. This extension shows that the primary principle of compressing one variable while preserving the information about another can be extended to handle any finite number of random variables. In particular, this extension defines a novel family of optimization problems, which are all special cases of one information-theoretic principle, the *multivariate* IB principle. We further show that analogous to the original IB problem, it is possible to characterize the form of the optimal solution to this multivariate principle. Additionally, we show how to extend all the algorithmic approaches suggested for the original IB problem, and apply the resulting algorithms to the analysis of a variety of real-world datasets. This part of the thesis is based on joint work with Nir Friedman, Ori Mosenzon and Naftali Tishby, and its preliminary versions were first presented in [32, 73].

Last, but not least, in Appendix A of this thesis we discuss the relationships of the IB framework to Maximum Likelihood of mixture models, which is a standard and well established approach to clustering. This appendix is based on joint work with Yair Weiss, and was first introduced in [79].

Abstract

This thesis introduces the first comprehensive review of the Information Bottleneck (IB) method along with its recent extension, the *multivariate* IB. The IB method was originally suggested in [82] as a new information-theoretic approach for data analysis. The basic idea is surprisingly simple: Given a joint distribution $p(x, y)$, find a compressed representation of X, denoted by T, that is as informative as possible about Y. This idea can be formulated as a variational principle of minimizing the mutual information, $I(T; X)$ (which controls the compactness of the representation T), under some constraint on the minimal level of mutual information that T preserves about Y, given by $I(T; Y)$. Hence, the fundamental trade-off between the complexity of the model and its precision is expressed here in an entirely symmetric form, where the exact same concept of information controls both its sides. Indeed, an equivalent posing of the IB principle would be to maximize the information T maintains about Y, where the (compression) information $I(T; X)$ is constrained to some maximal level.

As further shown in [82], this constrained optimization problem can be considered analogous to rate distortion theory, but with an important distinction: the distortion measure does not need to be defined in advance, but rather naturally emerges from the joint statistics, $p(x, y)$. Moreover, it leads to a tractable mathematical analysis which provides a formal characterization of the optimal solution to this problem. As an immediate implication, the IB method formulates a well defined information-theoretic framework for unsupervised clustering problems, which is the main focus of this thesis. Nonetheless, it is important to keep in mind that the same underlying principle of a trade-off between information terms may have further implications in other related fields, as recently suggested in [37].

After the introduction in Part I, in Part II we provide a detailed description of the IB method and its relations to rate distortion theory. We explicitly define some of the ideas that were briefly mentioned in [82], and further suggest new algorithmic approaches to construct solutions to the IB problem. Moreover, we provide empirical results that demonstrate the method's usefulness in a variety of applications, and discuss several related issues which are left for further research.

In Part III we suggest a general principled framework for multivariate extensions of the IB method. While the original principle suggested compressing one variable while preserving the information about another, this extension allows us to consider any finite number of random variables under the same framework. In particular, this extended formulation defines a novel family of optimization problems in which the original IB problem constitutes a special (important) case. These problems suggest novel approaches to data analysis, that to the best of our knowledge have not been treated or defined elsewhere. Specifically, we suggest considering multiple systems of data partitions that are interrelated, where Bayesian networks are utilized to specify the systems of clusters and which information terms should be maintained.

Analogous to the original IB problem, we characterize the form of the optimal solution to this general multivariate principle. That is, we are not satisfied with defining novel problems but also (formally) solve all of them at once. We further show how to extend all the algorithmic approaches suggested to the original IB problem, and apply the extended algorithms to solve three different IB variations with respect to several real world datasets. Nonetheless, we emphasize that additional applications of this multivariate framework still need be explored, and we expect that future research will elucidate such examples.

In the remainder of this abstract we provide a concise description of the chapters and appendices that constitute this thesis.

Chapter 1 forms a basic introduction to the remaining chapters. We start with a high level description of the fundamental precision-complexity trade-off and explain how the IB principle suggests a purely statistical and symmetric formulation to this trade-off. We further introduce the basic concepts that will be used throughout this thesis, and in particular the concepts of mutual and multi information.

In Chapter 2 we formally present the IB principle and its relationships to rate distortion theory. We define the concept of the relevance-compression function as a characteristic function for a given joint distribution, $p(x, y)$, and argue that this function can be considered as a natural extension to the well known rate-distortion

function. In the last section of this chapter we present the characterization of the optimal solution to the IB problem.

Chapter 3 describes four different and complementary algorithms that enable us to construct solutions in practice. The first two were originally suggested in [60, 82] while the other two are novel. We discuss the relationships between all these algorithms and suggest that combinations of these algorithms might also be useful in some cases.

In Chapter 4 we consider several applications of all the algorithms and combinations of algorithms suggested earlier. These applications further elucidate the earlier theoretical discussion, and demonstrate the applicability of the method to a variety of tasks. Due to the lack of space only a few applications are presented, while others are described in detail elsewhere [68, 75, 76, 77, 78, 83].

A preliminary assumption of the IB method is that the input is given in the form of a joint distribution. Nonetheless, in many situations this may not be the most natural representation. In Chapter 5 we investigate how to apply the IB framework to these situations as well, by applying a new pre-process procedure to the input data, termed here *Markovian relaxation*. We present additional applications to a variety of data types that demonstrate the effectiveness of combining this approach with the IB method.

Chapter 6 concludes the discussion regarding the original IB method. We present the relations between this method, as presented in this thesis, to recent related contributions [10, 19, 37], and point out several open problems and directions for further research.

In Chapter 7 we provide the necessary introduction to Part III. We motivate the search for a multivariate extension to the original IB framework and introduce the concept of Bayesian networks, which is the main tool we use in constructing this extension. We further relate this concept to the concept of multi-information through several simple propositions, which are necessary for the following analysis.

Chapter 8 describes the multivariate IB principle which provides a general extension to the original formulation. We further suggest an alternative and closely related variational principle that provides a different interpretation for the method. We discuss the relations between these two principles and demonstrate how to apply them in order to specify new (multivariate) IB-like variational problems.

In Chapter 9 we characterize the form of the optimal solution to the multivariate principles suggested in the previous chapter. In other words, we provide a general solution to all the optimization problems included in our multivariate framework. We demonstrate how this solution can be used as a "recipe" to induce concrete solutions to different specifications of multivariate IB problems.

In Chapter 10 we show how to extend all the four algorithmic approaches suggested for the original IB problem in order to solve multivariate IB constructions. Chapter 11 demonstrates several applications of the general methodology. Specifically, we apply all the extended algorithms to construct solutions to different multivariate IB problems with respect to a variety of real world datasets.

We conclude Part III in Chapter 12 where we discuss our results and some of their possible implications for future research.

Last, there are four appendices to this work. Appendix A provides a detailed discussion as regard to the relationships between the IB method and a well established probabilistic framework for clustering, known as Maximum Likelihood of mixture models. Although both approaches stem from conceptually different motivations, it turns out that in some cases there are some mathematical equivalences between them, as discussed in detail in this appendix. In Appendix B we provide a theoretical analysis that relates the concept of (relevant) mutual information with the supervised learning concept of precision. In particular, we show that under certain assumptions, seeking clustering solutions which are closer to the "true" partition of the input data is equivalent to seeking partitions that are more informative about the feature space of these data. Appendix C and Appendix D present the proofs for the theorems and propositions that are introduced throughout Part II and Part III, respectively.

Contents

Part I

General Background

Chapter 1

Introduction

In this chapter we provide a basic introduction to the remaining chapters. In the first section we present high level descriptions of the fundamental trade-off between precision and complexity. One important variant of this trade-off is formulated as the problem of unsupervised clustering, which is the main problem we address in this thesis. In the next section we present the necessary preliminaries for our analysis. We conclude this chapter by presenting a simple example in order to elucidate the central ideas that will be discussed later on.

1.1 The precision-complexity trade-off as a central paradigm

We start by briefly adapting the general description of a model of supervised learning, as given in page 17 in [86]. Such a model can be described as consisting of three components:

- A generator of random vectors $x \in \mathcal{R}^d$, drawn independently from an unknown probability distribution $p(x)$.
- A supervisor who returns a scalar output value $y \in \mathcal{R}$, according to an unknown conditional probability distribution $p(y \mid x)$.
- A learning machine capable of implementing a predefined set of functions, $f(x, \theta) : \mathcal{R}^d \times \Theta \to \mathcal{R}$, where Θ is a set of parameters.

The problem of learning is that of choosing from the given set of functions, the one that best approximates the supervisor's response. The choice is typically based on a training set of n independent and identically distributed pairs of observations drawn according to $p(x, y) = p(x)p(y \mid x)$:

$$\{(x_1, y_1), \dots, (x_n, y_n)\} \ . \tag{1.1}$$

The quality of the chosen function is estimated based on the (average) discrepancy between the "true" response y of the supervisor to some *new* input x and the "machine" response provided by $f(x, \theta)$ to the same input.

A classic trade-off in this scenario is between the quality, or the *precision* of the approximation of the given data versus the simplicity, or the class *complexity* of the approximating function. An illustration of this trade-off is given in Figure 1.1. In this example, if one tries to approximate the supervisor's response through a relatively complex function (a polynomial of a high degree), it typically *over-fits* the training data. That is, although the discrepancy between the true responses and the machine responses are minimized for the training examples, the generalization ability is limited and the predicted y value for new examples will be poor (left panel in the figure). On the other hand, if one approximates the response through an overly simple function (e.g., a polynomial of a low degree), again the predictions for new examples are of low quality

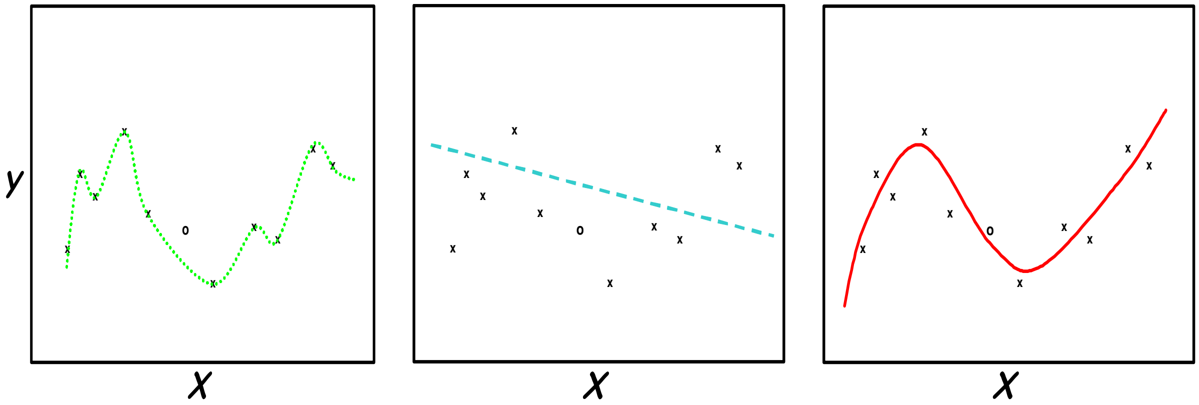

Figure 1.1: In all three panels, the horizontal axis corresponds to the x value while the vertical axis denotes the supervisor's response, y. Training examples are denoted by 'x', while a single test (new) example is denoted by 'o' (the same examples appear in all panels). **Left:** Approximating the training examples with a polynomial of a high degree typically over-fits these examples, and thus provides a poor prediction with respect to the new example. **Middle:** Approximating the training data with an overly simple function also provides poor test-set predictions. **Right:** Optimizing the trade-off between the complexity of the model (the degree of the approximating polynomial in this case) and the precision about the training examples typically yields good predictions with respect to new examples.

(middle panel in the figure). Therefore, a central goal is to obtain approximations which are simple enough on the one hand, and yet provide relatively precise approximations of the training data. In other words, one strikes a balance between the complexity of the model versus its precision about the training examples. The implicit assumption is that an approximation that optimizes this precision-complexity trade-off will be closer in nature to the real underlying process, which is formally expressed through $p(x, y)$. Hence, such an approximation is expected to minimize the prediction discrepancy with respect to *new* examples, as demonstrated in the right panel of the figure.

The optimization of this well known trade-off can be addressed in different ways. Common approaches include the Structural Risk Minimization (SRM) of Vapnik and Chervonenkis which stems from statistical learning theory considerations [86], Bayesian methods in which preference in favor of simpler models is implied through the prior (see, e.g., [13]), and Rissanen's Minimum Description Length principle [62] which is motivated by an information-theoretic analysis of the concept of randomness.

Although we introduced the precision-complexity trade-off in the context of supervised learning where "labels" (or supervisor's responses) are provided for the training examples, it is certainly prominent in the *unsupervised learning* scenario as well. Using the above notations, in this scenario one is given a set of *unlabeled* training examples, $\{x_1, \ldots, x_n\}, \; x_i \in \mathcal{R}^d$. Loosely speaking, the goal is to construct some compact representation of these data, which in some sense reveals their hidden structure. This representation can be used further to achieve a variety of goals, including reasoning, prediction, communication etc. (see, e.g., [51], Chapter 23). As implied by the somewhat vague phrasing of the two previous sentences, the definition of the problem of unsupervised learning, along with its goals, are less clear as compared to the supervised learning scheme. There are numerous different techniques for unsupervised data analysis, and comparing them is typically very difficult. Yet, one possible dichotomy splits unsupervised methods into *projection* [1] versus *clustering* methods.

In projection methods one aims to find a low-dimensional representation of the given high-dimensional data that preserves most of the "structure" contained in the original representation. Typically, some quality criterion is suggested, and in practice one tries to find a new low dimensional representation that at least

[1] The term "projection" is loosely used here. Specifically, we include in this category linear projection methods such as PCA, non-linear (continuous) dimensionality reduction methods such as SDR [37], and embedding techniques such as LLE [64].

locally optimizes this criterion. The most common technique is Principal Component Analysis (PCA), where one interpretation of this method states that it minimizes the squared distances from the original data points to their projections in the lower dimensional space.

In clustering methods, one is tackling the problem of unsupervised learning with a somewhat different approach. In its simplest form, a clustering solution is a partition of the input data into several exhaustive and mutually exclusive clusters. Each cluster can be represented by a centroid which is typically estimated as some weighted average of the cluster's members. [2] A "good" partition should group "similar" data points together, while "dissimilar" points are assigned to separate clusters. This implies that the quality of the partition can be estimated through the average distortion between the data points and their corresponding representatives (cluster centroids). In a more general formulation, first suggested in [60], each data point is assigned to *all* the clusters with some normalized probability. Thus, a clustering solution corresponds to a "soft" partition of the data points. In this case as well, the typical goal is to minimize the (weighted) average distortion between data points and cluster centroids.

Clearly, projection and clustering define deeply related tasks of dimensionality reduction. In fact, it is possible to formulate both approaches using very similar semantics, as done, e.g., in [48]. Nonetheless, these relationships are not relevant to the current discussion, hence we disregard them at this point. We henceforth concentrate on clustering methods, where first we are interested in exploring how the precision-complexity trade-off is expressed in this setting.

To this end, let us consider the illustrative example given in Figure 1.2. In this example the data points x_i are assumed to lie in $\mathcal{R}^2$. As in the supervised learning case, different models at different complexity levels can be suggested to cluster these data. For example, if we describe the data through two clusters (represented by dotted lines in the figure), we will have a rather compact model. However, at least for the right-hand cluster, the average distortion between data points and the cluster centroid will be relatively high. In our terminology this means that representing each data point through its corresponding cluster centroid will have poor precision. Thus, we might suggest a slightly more complex model which consists of three clusters. This can be done, e.g., by splitting the more scattered cluster into two more specific ones. Obviously, this more complex model will have better precision in the above sense. We may continue this line of thought, and think of more complex models that consist of additional clusters and provide better precision in their representation of the data. In the extreme case, each data point is assigned to a singleton cluster; thus, we have maximal precision since there is no discrepancy between data points and their representatives. Unfortunately, the description complexity is obviously maximized as well.

In information theory, which underlies most of the analysis presented in this thesis, this trade-off is treated through the sub-field of rate distortion theory. In particular, the complexity of the model is then characterized through its *coding length*, which in turn is proportional to the amount of (mutual) information between data points and their new representatives (precise definitions of all these concepts will be given shortly). If we term this information the "compression-information", simpler models correspond to models with low values of compression-information that enable more efficient communication. However, these models typically suffer from a relatively high (expected) distortion. Hence, this fast communication comes with the cost of lower precision of the sent messages. Thus, the familiar precision-complexity trade-off, which we already encountered for supervised and unsupervised learning, arises again in the context of communication through an information-theoretic analysis.

The tacit assumption of the above discussion is that a distortion measure between data points and cluster centroids is provided as part of the problem setup. Obviously, clustering algorithms as well as the estimated quality of their results crucially depend on the choice of the distortion measure. Unfortunately, to define such a measure is in many cases an extremely difficult task. As a result, this choice is (too) often an arbitrary

[2] For simplicity's sake we concentrate here on centroid based clustering techniques, also known as Vector Quantization algorithms. We disregard, for the moment, another important class of pairwise clustering methods in which clusters are not necessarily represented by centroids. We discuss the relations (in our context) between pairwise clustering and vector quantization in Chapter 5.

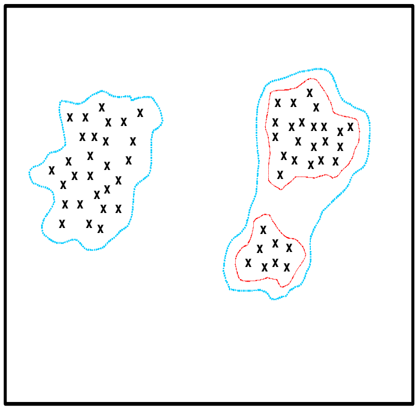

Figure 1.2: The precision-complexity trade-off in the context of unsupervised clustering. A simple model of two clusters will have low precision, in the sense that the distance (or distortion) between data points and their corresponding cluster centroid will be high. A more complex model of three clusters, where we split the more scattered cluster into two more specific ones, will have higher precision. That is, we can trade complexity with precision in a natural way.

one, which of course suppresses any *objective* (i.e., distortion independent) interpretation of the resulting clusters.

As we will see throughout this thesis, it is possible to cope with this potential pitfall in a well-defined way. More precisely, as first suggested in [82], the precision-complexity trade-off can be formulated without defining any distortion measure in advance. The basic idea is to use the exact same concept of mutual information in *both sides* of this trade-off. In particular, this is done by introducing a *relevant* variable, on which the mutual information should be preserved as high as possible, while the given data points are compressed. Denoting the information about this relevant variable as the “relevant information”, the precision-complexity trade-off is now formulated in an entirely symmetric form: we wish to minimize the compression-information while preserving the relevant information as high as possible. In this purely statistical formulation, complexity and precision are two sides of a *single* problem, as discussed in detail in the following chapters.

At first sight, this approach might look suspicious. In particular, it seems that we have replaced one problem of choosing an appropriate distortion measure with a new one of choosing the relevant variable. Although this statement is true, it turns out that in many practical situations the second problem is much easier to handle. Moreover, we argue that identifying the relevant variable is an important step in providing a more precise definition of the clustering task. In particular, it allows for a clear interpretation of the resulting clusters in terms of the compactness of the new representation versus the amount of information it preserves about the relevant variable. Furthermore, this formulation leads to a tractable mathematical analysis which is intimately related to the corresponding analysis described in rate distortion theory.

It is important to keep in mind that although this thesis concentrates on clustering problems, the underlying principle of a trade-off between two information terms, might have further implications in other related fields. In fact, recent work by Globerson and Tishby [37] have already shown that the same idea can be applied in the context of continuous dimensionality reduction methods, and a preliminary discussion of additional alternatives is given in [11]. Finally, in the above discussion we used the term “mutual information” without defining it precisely. In the next section we describe this definition along with the definitions of other related concepts.

1.2 Preliminaries

In this section we introduce the basic concepts required for the next chapters. We start with some notations, and further state the definitions of entropy, mutual and multi information, KL divergence and JS divergence. Most of this section is based on Chapter 2 in [20], Chapter 3 in [5] (which provides a friendly introduction to the concept of entropy), and a work in progress by Nemenman and Tishby [55], which introduces a new axiomatic derivative of mutual and multi-information.

1.2.1 Notations

Throughout this thesis we use the following notations. Capital letters $(X, Y, \ldots)$ denote the names of random variables. Lowercase letters $(x, y, \ldots)$ denote the realizations of the random variables, namely specific values taken by these variables. As a shortened notation we use $p(x)$ to denote $p(X = x)$; i.e., the probability that the assignment to the random variable X is the value x. We further use $X \sim p(x)$ to denote that X is distributed according to the probability distribution $p(x)$.

We use calligraphic notations, $(\mathcal{X}, \mathcal{Y}, \ldots)$ for the spaces to which the values of the random variables belong. Thus, $\mathcal{X}$ is the set of all possible values (or assignments) to X. The notation $\sum_x$ means summation over all $x \in \mathcal{X}$, and $|\mathcal{X}|$ stands for the cardinality of $\mathcal{X}$.

For simplicity, in this thesis we limit the discussion to discrete random variables with a finite number of possible values. That is, in our context, $(|\mathcal{X}|, |\mathcal{Y}|, \ldots)$ are all finite. Nonetheless, we emphasize that much of the following analysis can be extended to handle continuous random variables as well.

For Part III we need additional notations. We use boldface capital letters $(\mathbf{X}, \mathbf{Y}, \ldots)$ to denote sets of random variables. Specific values taken by those sets are denoted by boldface lowercase letters $(\mathbf{x}, \mathbf{y}, \ldots)$. The boldface calligraphic notation, $\boldsymbol{\mathcal{X}}$, denotes the set of all possible values to $\mathbf{X}$.

1.2.2 Entropy and related concepts

Consider the following situation. We are given a finite collection of documents, denoted by $\mathcal{Y} \equiv \{y_1, \ldots, y_{|\mathcal{Y}|}\}$. A person chooses to read a single document out of this collection, and our task is to guess which document was chosen. Without any prior knowledge, all guesses are equally likely. We now further assume that we have access to a definite set of (exhaustive and mutually exclusive) probabilities, denoted by $p(y),\ y \in \mathcal{Y}$, for all the possible choices. For example, let us assume that longer documents are more probable than shorter ones. More specifically, that the probability of choosing each document is proportional to the (known) number of words that occur in it. If all the documents consist of exactly the same number of words, $p(y)$ is uniform and obviously we are back at the starting point where no guess is preferable. However, if one document is much longer than all the others, $p(y)$ will have a clear peak for this document, hence our chances of providing the correct answer will improve. How can we *quantify* the difference between these two scenarios in a well defined way?

Loosely speaking, we may say that we are interested in quantifying the amount of "uncertainty" in a given probability distribution, $p(y)$. Thus, we need to seek for a functional that will provide a quantitative measure of the "uncertainty" associated with $p(y)$. Let us designate this functional by $H[p(y)]$. An alternative notation might be $H(Y)$ where Y is a random variable distributed according to $p(y)$. Importantly, though, H should depend *only* on $p(y)$, and not in any way on the correct value of Y (the chosen document in our example), nor on the possible values of Y (document identities in our case).

Shannon [70] suggested establishing such a measure by specifying several conditions (or *axioms*) that any such functional must satisfy. These conditions need to reflect our qualitative ideas about what a reasonable measure of uncertainty would be. Shannon defined three simple and most intuitive such conditions, and showed that there is only one mathematical functional that satisfies them. We now briefly review this classic derivation.

First, it is surely reasonable to require continuity. That is, we do not want infinitesimal changes in $p(y)$ to produce steep changes in the amount of uncertainty in $p(y)$. The first condition is thus:

Condition 1.2.1: $H(Y)$ should be continuous in $p(y)$.

Second, if all possible inferences are specified to be equally probable and we increase the number of inferences, it is intuitively acceptable that our uncertainty about the correct inference increases. In our example, if $p(y)$ is uniform, to guess what document was chosen is certainly easier for $|\mathcal{Y}| = 2$ than for $|\mathcal{Y}| = 3$. Therefore, the second condition is:

Condition 1.2.2: If $p(y) = \frac{1}{|\mathcal{Y}|}$ then $H(Y)$ should be a monotonically increasing function of $|\mathcal{Y}|$.

The third and last condition can be considered as a consistency requirement. We want the amount of uncertainty to be independent of the steps by which *certainty* may be achieved. Let us first work with a concrete simple example. We assume that there are three documents , where the corresponding probabilities are taken to be $p(y_1) = \frac{1}{2},\ p(y_2) = \frac{1}{3},\ p(y_3) = \frac{1}{6}$. Formally, the amount of uncertainty in this case can be expressed as $H(\frac{1}{2}, \frac{1}{3}, \frac{1}{6})$. However, we might think of a two-step process, where we group the inferences as $t_1 \equiv \{y_1, y_2\}$ and as $t_2 \equiv \{y_3\}$. We now first need to find out which is the correct group, where for this task the amount of uncertainty is expressed by $H(\frac{1}{2} + \frac{1}{3}, \frac{1}{6}) = H(\frac{5}{6}, \frac{1}{6})$. The second step is to determine the correct inference (i.e., document). If t_1 should happen to be the correct group, the remaining amount of uncertainty would be $H(\frac{3}{5}, \frac{2}{5})$. If the other group is the correct one the remaining uncertainty would be $H(1)$. Since all we know is the probability of each group to be correct, it seems reasonable to assess the amount of uncertainty in the second step by the weighted sum $\frac{5}{6}H(\frac{3}{5}, \frac{2}{5}) + \frac{1}{6}H(1)$. Thus, the total uncertainty in the two-step process is taken as the uncertainty needed to determine the group plus the weighted sum of the uncertainty needed to determine the correct inference, given the group. The consistency requirement states that the amount of uncertainty expressed in this way should agree with the amount of uncertainty expressed in the original one-step scheme. In our example, this means:

$$H(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}) = H(\frac{5}{6}, \frac{1}{6}) + \frac{5}{6}H(\frac{3}{5}, \frac{2}{5}) + \frac{1}{6}H(1) \ . \tag{1.2}$$

The generalization of this idea is formally expressed in the following condition:

Condition 1.2.3: For all possible groupings of $\mathcal{Y} = \{y_1, \ldots, y_{|\mathcal{Y}|}\}$ into the groups $\mathcal{T} \equiv \{t_1, \ldots, t_{|\mathcal{T}|}\}$, $t \subset \mathcal{Y}$, the function $H(Y)$ should satisfy the consistency relation:

$$H(Y) = H[p(y)] = H[p(t)] + \sum_t p(t) H[p(y \mid t)] \ . \tag{1.3}$$

The remarkable result is that these three simple conditions are sufficiently restrictive so that the mathematical function $H(Y)$ follows unambiguously. Specifically, the celebrated Boltzmann-Shannon entropy, given in the following definition, is the *only* function that satisfies the above three requirements.

Definition 1.2.4: Let Y be a discrete random variable distributed according to $p(y)$. The *entropy* of Y is defined by

$$H(Y) \equiv H[p(y)] = -\sum_y p(y) \log p(y) \ . \tag{1.4}$$

This function is defined up to a multiplicative constant, the base of the logarithm, that merely sets the scale for this measure. If the logarithm is chosen to be to the base 2, the entropy is expressed in bits. In this case it has the appealing interpretation as the (expected) minimal number of 'yes'/'no' questions required to determine the value of Y (in the following, though, we typically use the natural logarithm). Additionally, the

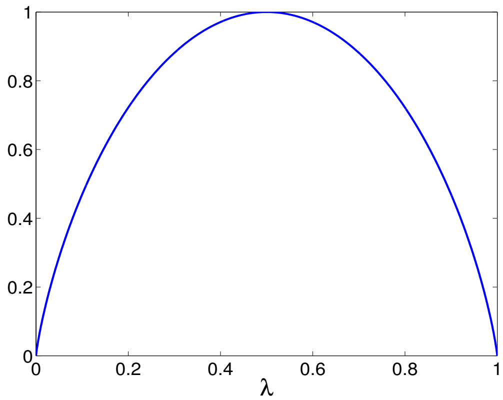

Figure 1.3: The entropy $H(Y)$ where Y has two possible values and $p(y_1) = \lambda$ (the base of the logarithm is 2 in this example). The entropy has a unique maximum for the uniform distribution ($\lambda = 0.5$) and it tends to decrease as $p(y)$ becomes less balanced.

entropy arises as the answer to several natural questions, such as "what is the average length of the shortest description of the random variable?"

Some immediate consequences of Definition 1.2.4 are given in the following proposition (see [20] for the proofs).

Proposition 1.2.5: $0 \leq H(Y) \leq \log |\mathcal{Y}|$ *and it is a concave function of* $p(y)$.

As a simple example consider the case where $Y = y_1$ with probability λ and $Y = y_2$ with probability $1 - \lambda$. In this case it is easy to verify that $H(Y) = 0$ if and only if $\lambda = 0$ or $\lambda = 1$. This coincides with our understanding that for $\lambda = 0$ or $\lambda = 1$, the variable Y is not random and there is no uncertainty. On the other hand, $H(Y)$ has a unique maximum for $\lambda = \frac{1}{2}$, which also corresponds to our intuition that in this case the uncertainty about the value of Y is maximized. Moreover, in Figure 1.3 we see that $H(Y)$ is continuous in λ, as implied by Condition 1.2.1, and in particular that $H(Y)$ tends to decrease as the underlying distribution becomes less balanced. This (somewhat loose) observation is true in the general case as well where $|\mathcal{Y}| > 2$.

We now extend the entropy definition to a set of random variables, $\mathbf{Y} \equiv \{Y_1, \ldots, Y_n\}$. Since obviously $\mathbf{Y}$ is simply a single vector-valued random variable, there is nothing new in this definition.

Definition 1.2.6: Let $\mathbf{Y} \equiv \{Y_1, \ldots, Y_n\}$ be a set of n discrete random variables distributed according to $p(y_1, \ldots, y_n)$. The *joint entropy* of this set is defined as

$$H(Y_1, \ldots, Y_n) = - \sum_{y_1, \ldots, y_n} p(y_1, \ldots, y_n) \log p(y_1, \ldots, y_n) \ . \tag{1.5}$$

In particular, if we have only two random variables, X and Y, we obtain $H(X, Y) = - \sum_{x,y} p(x, y) \log p(x, y)$. Additionally, the conditional entropy of a random variable given another is defined through the following definition.

Definition 1.2.7: If $(X, Y) \sim p(x, y)$, then the *conditional entropy* of Y given X is defined as

$$H(Y \mid X) = \sum_x p(x) H(Y \mid X = x) = - \sum_x p(x) \sum_y p(y \mid x) \log p(y \mid x) \ . \tag{1.6}$$

Expressed in words, $H(Y \mid X)$ is the (expected) uncertainty remaining on Y once we know the value of X. In the following subsection we show how these definitions are related to the concept of mutual and multi information, which are the fundamental concepts we deal with throughout this thesis.

1.2.3 Mutual information and multi-information

Let us reconsider our previous example of trying to guess what document was chosen. However, we now assume that we have access not only to the prior distribution $p(y)$, but rather to a *joint* distribution of Y with some other random variable, X. For concreteness, if Y values correspond to all the possible document identities, let us assume that X values correspond to all the distinct *words* occurring in this document collection. Thus, more formally stated, we assume that we have access to the joint distribution $p(x, y)$ which indicates the probability that a random word position in the corpus is equal to $x \in \mathcal{X}$ while the document identity is $y \in \mathcal{Y}$. [3]

We now further assume that after the document is chosen, we are informed about some of its contents, that is about some words that occur in it. Clearly, these details, accompanied by the knowledge of $p(x, y)$, improve our chances. For example, assume that there is some specific word x_i that occurs *only* in the document y_j. If we are lucky enough to have this word among the ones that we are told about, the game is over and we have full certainty that the chosen document was y_j. Hence, while we try to predict the value of Y (which was sampled according to $p(y)$), knowing the (sampled) values of some correlated variable, X provides some guidance, which we may fairly term as the "information" that X provides about Y.

As in the entropy case, a natural desire is to *quantify* how much "information" X contains about Y. Shannon already addressed this issue through his well known definition of mutual information.

Definition 1.2.8 : Let (X, Y) be two discrete random variables, distributed according to $p(x, y)$ and with marginal distributions $p(x) = \sum_y p(x, y)$ and $p(y) = \sum_x p(x, y)$. The *mutual information* between X and Y is defined as

$$I(X;Y) \equiv I[p(x,y)] = \sum_x \sum_y p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \ . \tag{1.7}$$

Using this definition and the above mentioned definitions, it is easy to obtain

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \tag{1.8}$$

$$I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \ . \tag{1.9}$$

These relations are commonly expressed in a Venn diagram as in Figure 1.4 (which is reproduced here from [20]). In particular, these relations suggest insightful interpretations as to the concept of mutual information. For example, since the entropy is a lower bound on the minimal achievable code length for the corresponding random variable, Eq. (1.8) implies that $I(X;Y)$ measures the average number of bits (if the logarithm is in base 2) that can be gained by a joint compression of X and Y versus independent compression that ignores their possible correlations. Alternatively, Eq. (1.9) states that the mutual information is the reduction in the uncertainty of Y due to the knowledge of X. In particular, if we continue using logarithms to the base 2, we may say that $I(X;Y)$ corresponds to the (expected) number of 'yes'/'no' questions one should ask one of the variables in order to learn all that it knows about the other [55]. In the extreme case where knowing each value of X provides complete knowledge of the value of Y (i.e., Y is deterministic given any $x \in \mathcal{X}$), the information between X and Y is maximized, or equivalently the reduction in the uncertainty about Y due to the knowledge of X is maximized.

As in the case of the entropy definition, the mutual information given by Definition 1.2.8 turns out to be the natural answer to many fundamental questions in information theory. Perhaps the two most well known results are the channel coding theorem and the rate distortion theorem which we discuss later on. In particular, $I(X;Y)$ characterizes the (expected) maximal number of bits that can be reliably sent in a (discrete memoryless) channel with a probability transition matrix $p(y \mid x)$.

[3] For brevity we take the simplifying "bag of words" assumption, which implies that the *order* of the words in each document has no effect on this distribution. More specifically, we may assume that $p(x, y)$ is given by the number of occurrences of the word x in the document y, normalized by the total number of words in the corpus.

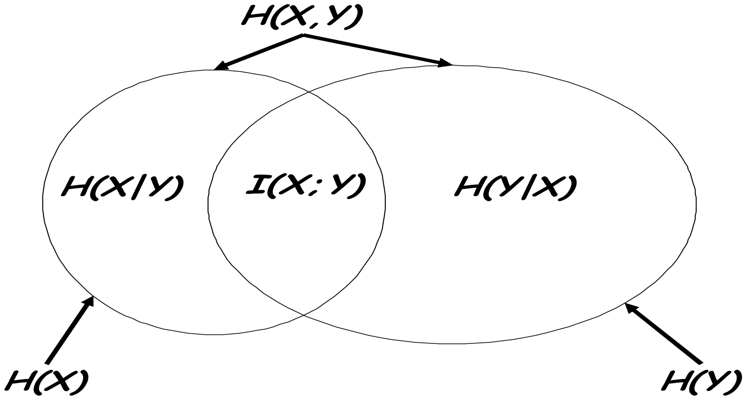

Figure 1.4: Relations between entropy, joint entropy, conditional entropy and mutual information for two random variables.

Nonetheless, somewhat surprisingly, an axiomatic derivation of this concept (as was done for entropy by Shannon), was introduced only recently by Nemenman and Tishby [55]. Specifically, they suggested a natural and intuitive set of conditions (or axioms) that should reflect our qualitative notion of the concept of "information" between random variables. The first three are natural extensions to the conditions suggested for the entropy concept, while the fourth condition is a simple symmetry requirement. [4] We now briefly review this derivation.

First, we want to ensure that small changes in $p(x, y)$ will not produce abrupt changes in the information.

Condition 1.2.9: $I(X;Y)$ should be continuous in $p(x, y)$.

Second, let us assume that choosing a value for X or Y defines the other uniquely, and additionally $p(x)$ and $p(y)$ are uniform and $k \equiv |\mathcal{X}| = |\mathcal{Y}|$. In this situation, it is reasonable to require that the information between X and Y will increase with k. This gives rise to the second condition.

Condition 1.2.10: If $p(x) = p(y) = \frac{1}{k}$ and choosing some value in X or in Y determines the other value uniquely, then $I(X;Y)$ should be a monotonically increasing function of k.

Further, the entropy consistency requirement is also easily extended to this context. Again, we want the amount of information to be independent of the steps by which this information is provided. This is formally expressed in the next condition.

Condition 1.2.11: For all possible groupings of $\mathcal{X} = \{x_1, x_2, \ldots, x_{|\mathcal{X}|}\}$ into the groups $\mathcal{T} \equiv \{t_1, \ldots, t_{|\mathcal{T}|}\}$, $t_k \subset \mathcal{X}$, [5] the function $I(X;Y)$ should satisfy the consistency relation:

$$I(X;Y) = I[p(x,y)] = I[p(t,y)] + \sum_t p(t) I[p(x, y \mid t)] \ . \qquad (1.10)$$

Last, it seems intuitively reasonable to ask for symmetry, i.e., that the information X provides about Y will be equal to the information Y provides about X.

Condition 1.2.12: The information should be symmetric in its arguments:

$$I(X;Y) = I(Y;X) \ . \qquad (1.11)$$

[4] An open question is whether this set is the *minimal* set required such that the information will be defined unambiguously. This issue is not addressed in [55].

[5] Note that while in the previous section T denoted a partition of Y values, henceforth it denotes a partition of X values.

As shown in [55], these four conditions suffice so that the mathematical function of information follows unambiguously. In particular, the mutual information as defined in Definition 1.2.8 is the *only* function that satisfies the above conditions.

To summarize, we saw that an axiomatic derivative for the concept of information between two random variables is possible. An immediate question is whether this concept can be extended to quantify the shared information between more than two random variables. A possible, and rather natural extension has been suggested over the years (see, e.g., [80], and the references therein), and is described in the following definition.

Definition 1.2.13 : Let $(X_1, \ldots, X_n)$ be a set of n discrete random variables, distributed according to $p(x_1, \ldots, x_n)$ and with marginal distributions $p(x_i) = \sum_{x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n} p(x_1, \ldots, x_n)$. The *multi-information* between these n variables is defined as

$$\mathcal{I}(X_1, \ldots, X_n) \equiv \mathcal{I}[p(x_1, \ldots, x_n)] = \sum_{x_1,\ldots,x_n} p(x_1, \ldots, x_n) \log \frac{p(x_1, \ldots, x_n)}{p(x_1) \ldots p(x_n)} . \tag{1.12}$$

Clearly, for $n = 2$ we are back in the standard pairwise concept of mutual information. The multi-information captures how close the distribution $p(x_1, \ldots, x_n)$ is to the factored distribution of the marginals. If this quantity is small, we do not lose much by approximating $p(x_1, \ldots, x_n)$ through the product distribution. Alternatively, as in the mutual information case, it measures the average number of bits that can be gained by a joint compression of the variables versus independent compression. The relations between entropy and mutual information are also easily extended to the multi-information case. For example (see [55])

$$\mathcal{I}(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i) - H(X_1, \ldots, X_n) , \tag{1.13}$$

which is the multivariate analogous to Eq. (1.8).

Naturally, we would like to provide some axiomatic derivative to this definition as well. To achieve this goal Nemenman and Tishby showed that only one "inductive" condition should be added to the previous four. This condition states that if a new variable is added, the multi-information increases exactly by the amount of information between the new variable and its preceding ones. This requirement is expressed in the next condition.

Condition 1.2.14: The multi-information should satisfy

$$\mathcal{I}(X_1, \ldots, X_{n+1}) = \mathcal{I}(X_1, \ldots, X_n) + I(X_1, \ldots, X_n; X_{n+1}) . \tag{1.14}$$

Note that the second term in the right-hand side is the *mutual* information between the vector-valued random variable $X_1 \times \ldots \times X_n$ and X_{n+1}. As shown in [55], the only function that satisfies the five conditions presented in this section is the multi-information as defined in Definition 1.2.13. Clearly, as $p(x_1, \ldots, x_n)$ becomes "more similar" to $p(x_1) \ldots p(x_n)$, the multi-information drops accordingly. In the next section, this relation is formally established.

1.2.4 KL divergence

Definition 1.2.15: The *Kullback Leibler (KL) divergence* between two probability distributions $p_1(x)$ and $p_2(x)$ is defined as

$$D_{KL}[p_1 \| p_2] = \sum_x p_1(x) \log \frac{p_1(x)}{p_2(x)} , \tag{1.15}$$

where the limits $0 \log \frac{0}{p_2} \to 0,\ p_1 \log \frac{p_1}{0} \to \infty$ are implied.

This divergence measure, also known as the *relative entropy* between $p_1(x)$ and $p_2(x)$, measures the "distance" between its two arguments. Note, though, that it is certainly not a metric distance since it is not symmetric and does not satisfy the triangle inequality. The KL arises in many fields as a natural divergence measure between two distributions. In particular, it quantifies the coding inefficiency of assuming that the distribution is $p_2(x)$ when the true distribution is $p_1(x)$. Specifically, if we have access to the true distribution $p_1(x)$, we can then construct a code for X with an average description length of $H(X) = H[p_1(x)]$. If, instead, we use the code for a distribution $p_2(x)$, we would need $H[p_1(x)] + D_{KL}[p_1(x) \| p_2(x)]$ bits on the average to describe the random variable (see [20], page 18).

Using this definition and the definitions of the previous section, it is easy to verify that

$$\mathcal{I}(X_1, \ldots, X_n) = D_{KL}[p(x_1, \ldots, x_n) \| p(x_1) \ldots p(x_n)] \ . \tag{1.16}$$

That is, the multi-information is the KL divergence between the joint distribution and the factored distribution of the marginals. In particular, for the pairwise mutual information we have

$$I(X;Y) = D_{KL}[p(x,y) \| p(x)p(y)] \ . \tag{1.17}$$

The following inequality, sometimes referred to as *Information inequality*, states that the KL divergence is non-negative and equals zero if and only if $p_1 = p_2$ (see [20], page 26, for the proof).

Proposition 1.2.16: *Let $p_1(x)$ and $p_2(x)$ be two probability distributions, then*

$$D_{KL}[p_1(x) \| p_2(x)] \geq 0 \tag{1.18}$$

with equality if and only if $p_1(x) = p_2(x), \ \forall x \in \mathcal{X}$.

Thus, using the last three equations we have that the multi-information and the mutual information are always non-negative, which agrees with the intuition about the required properties of an information measure.

The KL is not the only possible divergence measure between probability distributions. One alternative, which is especially important in our context, is presented in the next section.

1.2.5 JS divergence

Definition 1.2.17: The *Jensen-Shannon (JS) divergence* between two probability distributions $p_1(x)$ and $p_2(x)$ is defined as

$$JS_\Pi[p_1, p_2] = \pi_1 D_{KL}[p_1 \| \bar{p}] + \pi_2 D_{KL}[p_2 \| \bar{p}] \ , \tag{1.19}$$

where $\Pi = \{\pi_1, \pi_2\}, \ 0 < \pi_1, \pi_2 < 1, \ \pi_1 + \pi_2 = 1$ and $\bar{p} = \pi_1 p_1 + \pi_2 p_2$.

This divergence measure was first introduced in this form in [29]. However, it was first defined under this name by Lin [50], and it appeared earlier in the literature under additional different names (see, e.g., [88] where it was termed the increment of Shannon entropy). Lin used a somewhat different form, given by:

$$JS_\Pi[p_1, p_2] = H[\bar{p}] - \pi_1 H[p_1] - \pi_2 H[p_2] \ , \tag{1.20}$$

where Π and $\bar{p}$ are defined as in Definition 1.2.17 and $H[p]$ is Shannon entropy. Simple algebra can show that Eq. (1.19) and Eq. (1.20) are equivalent.

The JS measure also has coding-theoretical motivation as thoroughly discussed in [69]. Using Proposition 1.2.16 it follows that it is non-negative and equals zero if and only if $p_1 = p_2$. It is also symmetric in p_1 and p_2, but it does not satisfy the triangle inequality, hence it is not a metric.

As shown by Gutman [42] and further discussed in [69], the JS measure is tightly related to the known two-sample problem [49]. In its general formulation, the two-sample problem is to decide whether two samples independently drawn from two unknown distributions in a predefined family, are actually drawn from the same distribution. Gutman showed that comparing the JS divergence between the two corresponding *empirical* distributions (or *types*) with some predefined threshold, is (asymptotically) the optimal test for this task. The optimality in this result is in the Neyman-Pearson sense (see [20], page 305), and the only assumption is that the underlying family of distributions is the class of stationary and ergodic Markov sources.

Another observation which is important in our context relates the JS divergence to the concept of mutual information. In particular, Lin already suggested in [50] to extend Eq. (1.20) to measure the divergence between more than two distributions, through:

$$JS_{\Pi}[p_1, \ldots, p_n] = H[\bar{p}] - \sum_{i=1}^{n} \pi_i H[p_i] \ , \tag{1.21}$$

where $\Pi = \{\pi_1, \ldots, \pi_n\}, \ 0 < \pi_i < 1, \ \sum_{i=1}^{n} \pi_i = 1$ and $\bar{p} = \sum_{i=1}^{n} \pi_i \ p_i$. For a given joint distribution $p(x, y)$, using the notations $\pi_i = p(x_i), \ p_i = p(y \mid x_i), \ n = |\mathcal{X}|$, then clearly $p(y) = \sum_x p(x) \ p(y \mid x) = \sum_{i=1}^{n} \pi_i \ p_i = \bar{p}$. Using Eq. (1.9) we thus find

$$I(X;Y) = H[p(y)] - \sum_x p(x) H[p(y \mid x)] = H[\bar{p}] - \sum_{i=1}^{n} \pi_i H[p_i] = JS_{\Pi}[p_1, \ldots, p_n] \ . \tag{1.22}$$

In other words, if we take the weights in Π as the prior probabilities $p(x)$, the mutual information between X and Y is exactly the JS divergence between all the conditional distributions, $p(y \mid x)$.

To gain some intuition into this equivalence we again consider the example of documents and words. If all the conditional word distributions are the same (i.e., all documents have similar relative word frequencies) the JS is clearly zero. Accordingly, in this case there is no information between X and Y and knowing what words are present in the chosen document will be useless. On the other hand, if the conditional word distributions are very different from each other (i.e., different documents typically consist of different words) the JS will be relatively high. Accordingly, in this case there is a lot of information between X and Y, and knowing the words in the chosen document will significantly improve our chances of guessing its identity.

An immediate corollary of Eq. (1.22) is that the JS is a bounded divergence measure (since the mutual information is always bounded). This is in contrast to the KL divergence, which is not bounded in the general case and in particular highly sensitive to low probability values in its second argument. Additionally, the mutual information is known to be a concave function of $p(x)$ for fixed $p(y \mid x)$ and a convex function of $p(y \mid x)$ for fixed $p(x)$ (see [20], page 31). Thus, using Eq. (1.22) we see that for fixed $p_1, \ldots, p_n$ the JS is a concave function of Π. For the pairwise JS measure, this means that for fixed $p_1 \neq p_2$ the JS approaches zero when $\pi_1 \to 0$ or $\pi_1 \to 1$, and reach its unique maximal value for $\pi_1 = \pi_2 = \frac{1}{2}$.

1.3 Relevant versus irrelevant distinctions

To end this introduction, let us reconsider one last variant of our game of guessing the chosen document identity. Up to now we have assumed that we have access to the joint distribution of documents and words, denoted by $p(x, y)$. We now further assume that before making our guess we are allowed to get answers to a *limited* number of binary questions regarding the document content. Specifically, possible questions must be in the form of 'does the word x_i appear (does not appear) in the document?' or 'do any of the words in t_k appear (do not appear) in the document?' (where $t_k \subset \mathcal{X}$ is some subset of words).

Since the number of questions is limited we are obviously interested in asking the "most informative" ones. While the mutual information can be viewed as the expected *number* of binary question we need to ask about X in order to learn about Y, it provides no guidance whatsoever in our current context. To put it bluntly, this concept does not tell us anything as regards *what* questions we need to ask.

Table 1.1: A simple example where it is possible to partition X values into a small number of clusters such that in each cluster all the conditional distributions, $p(y \mid x)$ are identical. In this case, in order to preserve all the information contained in X about Y, one needs only to preserve the distinctions between the clusters of X values.

T	X	$p(y_1 \mid x)$	$p(y_2 \mid x)$	$p(y_3 \mid x)$
t_1	x_1	0.9	0.1	0.0
	x_2	0.9	0.1	0.0
	x_3	0.9	0.1	0.0
	x_4	0.9	0.1	0.0
t_2	x_5	0.2	0.1	0.7
	x_6	0.2	0.1	0.7
t_3	x_7	0.0	0.5	0.5
	x_8	0.0	0.5	0.5

Nevertheless, it is intuitively clear that not all questions are equally helpful. For example, if some word x_i occurs exactly once in every document, asking about this specific word is clearly useless. On the other hand, if x_i occurs in exactly half of the documents, asking about it seems useful since the answer will substantially reduce the number of alternatives (documents) between which we need to choose.

Additionally, since we are limited in the number of questions, in general it seems desirable to ask about groups, or *clusters* of words, rather than about specific ones. As a simple example, let us assume that x_i and $x_{i'}$ always occur together, which formally means that their conditional document distributions are identical, i.e., $p(y \mid x_i) = p(y \mid x_{i'})$. In this case, if we get a positive (negative) answer while we ask about x_i, there is no point to further ask about $x_{i'}$ since we surely know that the answer will be positive (negative) as well. Thus, it seems reasonable to treat these two words as a single "feature" and ask whether *any* of the words in $t = \{x_i, x_{i'}\}$ occurs in the chosen document.

Extending this idea, let us assume that there are several clusters of words, denoted as $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$, where in each such cluster all the conditional document distributions are identical. That is, $x_i, x_{i'} \in t_k$ if and only if $p(y \mid x_i) = p(y \mid x_{i'})$. Obviously, we can always find such a partition if we do not limit the number of clusters. However, for the purposes of this discussion let us assume that there exists such a partition with $|\mathcal{T}| \ll |\mathcal{X}|$. This situation is demonstrated in Table 1.1. Obviously, in our specific task of predicting the chosen value of Y, *the distinctions inside each such cluster of words are irrelevant*. Hence, detecting these irrelevant distinctions should yield an optimal set of questions that focus solely on the presence or absence of the corresponding word *clusters*, rather than on the words directly.

The above example seems a bit forced. Nonetheless, the same intuition holds even if we relax our previous requirements. That is, we simply assume that words are assigned to the same cluster if their conditional document distributions are "similar" in some sense. This situation is demonstrated in Table 1.2. Here, again, several distinctions in X seems redundant, while others are highly relevant and informative about Y. In particular, to inspect the corresponding word clusters seems like a useful way to extract most of the information about Y through a minimal set of questions.

While the above arguments might be intuitively acceptable, real-life examples are obviously more complicated. In particular, there is an exponential number of possible partitions of X values. Thus, some guiding principle for choosing the "best" partition can provide much assistance. Recall that Condition 1.2.11 in the axiomatic derivative of the mutual information states that the amount of information between X and Y is independent of the steps (or questions) by which this information is provided. In particular, if we reconsider Eq. (1.10) we see that the information can be viewed as a sum of two terms:

$$I(X;Y) = I[p(x,y)] = I[p(t,y)] + \sum_t p(t) I[p(x,y \mid t)] \, , \tag{1.23}$$

Table 1.2: A relaxed variant of the example in Table 1.1. Even if the conditional distributions, $p(y \mid x)$ in each cluster $t \in \mathcal{T}$ are just "similar" and not identical, the clusters preserve most of the information that X contains about Y.

T	X	$p(y_1 \mid x)$	$p(y_2 \mid x)$	$p(y_3 \mid x)$
t_1	x_1	0.91	0.08	0.01
	x_2	0.89	0.09	0.02
	x_3	0.88	0.11	0.01
	x_4	0.93	0.05	0.02
t_2	x_5	0.18	0.08	0.74
	x_6	0.16	0.12	0.72
t_3	x_7	0.02	0.47	0.51
	x_8	0.03	0.51	0.46

where $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$ defines a partition of $\mathcal{X}$ into exhaustive and mutually exclusive groups (or clusters). It turns out that our intuitive hints a few lines ago are directly related to this presentation. More specifically, a "good" partition of $\mathcal{X}$, where words with "similar" ("non-similar") conditional distributions over the documents are grouped together (apart), corresponds to a high $I[p(t, y)]$ value. This result which might looks a bit vague at this point, is in the core of this thesis and thus will be discussed in detail later on.

In the above discussion, an implicit assumption is that there is some meaningful way to measure the "similarity" between conditional distributions. However, in the previous section we saw two different divergence measures between probability distributions, and many more exist but are not mentioned here. Which measure is preferable? A central and somewhat surprising result is that this question can be disregarded. More precisely, if we accept that our guiding principle is to seek for partitions of X values that are highly informative about Y, there is no need to define some similarity or distance measure in advance. In particular, the Information Bottleneck method provides a mathematical formulation for such a principle. Furthermore, mathematical analysis of this principle serves to characterize the form of the *optimal* partitions of X values that maximize the information about the value of Y. These results are the topic of the next part of this thesis.

Part II

The Single Sided Information Bottleneck

Chapter 2

The IB Variational Principle

We start this chapter by a brief overview of rate distortion theory which is mainly based on the corresponding sections in [20, 82]. After this introduction we describe the IB variational principle and show that in several respects, it can be considered as an extension of rate distortion theory. The last section of this chapter characterizes the (formal) optimal solution to the IB principle.

2.1 Brief overview of rate distortion theory

Let X be a discrete random variable with a finite set of possible values, $\mathcal{X}$, distributed according to $p(x)$. As the cardinality $|\mathcal{X}|$ increases, a perfect representation of this random variables becomes more demanding. However, as we will see later on, a perfect representation might be redundant and unnecessary, depending on the task at hand.

Let T denote some other discrete random variable which is a compressed representation (or quantized codebook) of X. This representation is defined through a (possibly stochastic) mapping between each value $x \in \mathcal{X}$ to each representative value $t \in \mathcal{T}$. Formally, this mapping can be characterized by a conditional distribution $p(t \mid x)$, inducing a soft partitioning of X values. Specifically, each value of X is associated with all the codebook elements (T values), with some normalized probability.

What determines the quality of this compressed representation? The first factor is obviously how compressed it is. A standard measure for this quantity is the *rate* of a *code* with respect to a *channel* "transmitting" between X and T. An exact definition of these concepts is not necessary for our needs and we can be satisfied with a strongly related quantity given by the mutual information $I(T;X)$, which we will term the *compression-information*. Note that this information is calculated based on the joint distribution $p(x)p(t \mid x)$. Low values of this quantity imply more compact representations. For example, in the extreme case where T has just a single value, clearly $I(T;X) = 0$. On the other hand, redundant representations imply high compression-information values. For example, if T simply copies X (i.e., no compression), we have $I(T;X) = H(X)$ which is the upper bound of this term.

A more formal interpretation relates $I(T;X)$ to the maximal number of bits that can be reliably transmitted from X to T. In the following we provide some intuition on this result. In principle, a reliable transmission requires that different input sequences will produce disjoint output sequences. Using the Asymptotic Equipartition Property (AEP) [20], it is possible to see that for each (typical) n-sequence of T symbols, there are $\approx 2^{nH(X|T)}$ possible X ("input") n-sequences, all of them are equally likely. Using again AEP we see that the total number of (typical) X n-sequences is $\approx 2^{nH(X)}$. We need to ensure that no two X sequences will "produce" the same T sequence. Hence, the set of possible X sequences has to be divided into subsets of size $2^{nH(X|T)}$, where each subset corresponds to (or is "clustered into") some different T n-sequence. The total number of such disjoint subsets is upper bounded by $2^{n(H(X)-H(X|T))} = 2^{nI(T;X)}$. Therefore, we can send at most $\approx 2^{nI(T;X)}$ distinguishable sequences of length n from X to T. In Figure 2.1

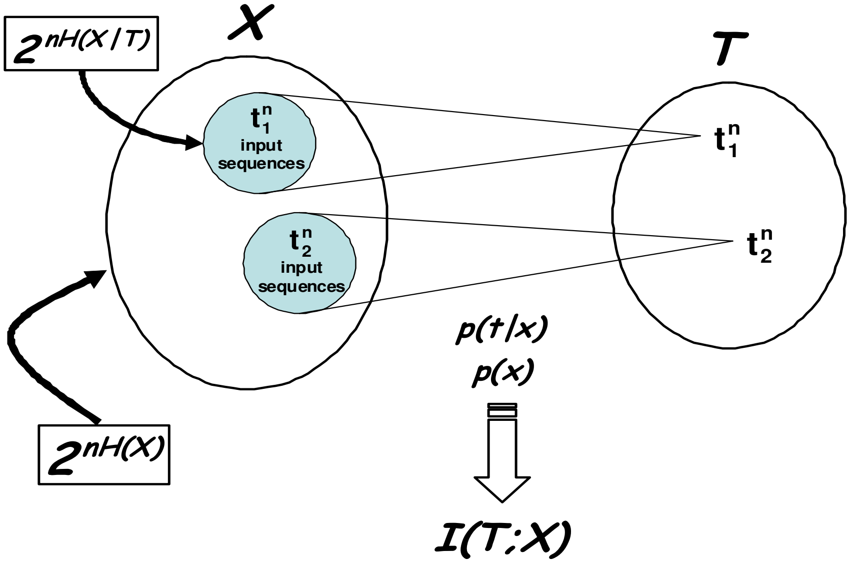

Figure 2.1: An illustration of the relation between the compression-information, $I(T;X)$, and the maximal number of bits that can be reliably transmitted between X and T. For every typical sequence of length n of T symbols there are $\approx 2^{nH(X|T)}$ possible ("input") X sequences. Hence, the total number of $\approx 2^{nH(X)}$ X sequences needs to be divided into disjoint subsets of size $\approx 2^{nH(X|T)}$. The number of such subsets is therefore upper bounded by $2^{n(H(X)-H(X|T))} = 2^{nI(T;X)}$. In other words, we can reliably send at most $\approx 2^{nI(T;X)}$ sequences of length n between X and T.

we illustrate this idea.

Summarizing the above arguments we see that $I(T;X)$ measures the compactness of the new representation, T. However, this quantity alone is not enough. Clearly the compression-information can always be reduced by ignoring further details in X (e.g., by using only a single value in T). Therefore, additional constraints are needed. In rate distortion theory this is accomplished by defining a *distortion measure* which measures the "distance" between the random variable and its new representation. Specifically, a function $d : \mathcal{X} \times \mathcal{T} \to \mathcal{R}^+$ must be defined to complete the setup of the problem, where the assumption is that smaller distortion values imply a better representation. Given such a function, the partitioning of $\mathcal{X}$ induced by $p(t \mid x)$ has an *expected distortion* of:

$$\langle\, d(x,t)\, \rangle_{p(x)p(t|x)} = \sum_{x,t} p(x)p(t \mid x)d(x,t)\ . \tag{2.1}$$

The trade-off between the compactness of the new representation and its expected distortion is the fundamental trade-off in rate distortion theory. This trade-off was first characterized by Shannon through the rate-distortion function, which is discussed in the next section.

2.1.1 The rate-distortion function

The rate-distortion function, denoted by $R(D)$, is defined given the source statistics, $p(x)$ and some distortion measure $d(x,t),\ \forall x \in \mathcal{X},\ \forall t\ \in \mathcal{T}$. The "operational" definition defines $R(D)$ as the infimum of all rates R under a given constraint on the average distortion D. An alternative mathematical definition is given by

$$R(D) \equiv \min_{\{p(t|x):\ \langle\, d(x,t)\, \rangle \leq D\}} I(T;X)\ . \tag{2.2}$$

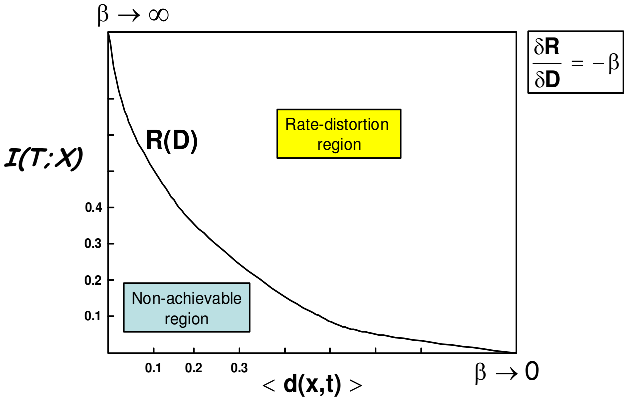

Figure 2.2: An illustration of a rate distortion function, $R(D)$. This function defines a monotonic convex curve in the distortion-compression plane with a slope of $-\beta$. When $\beta \to \infty$ we focus solely on minimizing the distortion which corresponds to the extreme case of the curve with $\langle\, d(x,t)\, \rangle_{p(x)p(t|x)} \to 0$. When $\beta \to 0$ we are only interested in compression, which corresponds to the other extreme of the curve with $R \to 0$. This curve characterizes the input (source) statistics, $p(x)$ with respect to a specific distortion measure and a specific choice of representatives, given by T values. The region above the curve is achievable while the region below it is non-achievable.

In other words, $R(D)$ is the minimal achievable compression-information, where the minimization is over all the normalized conditional distributions, $p(t \mid x)$ for which the distortion constraint is satisfied.

The first main result of rate distortion theory, due to Shannon is that these two definitions are equivalent (see, e.g., [20], page 342). Thus, for our needs we will concentrate on this second definition.

The trade-off characterized by $R(D)$ is monotonic: higher D values (i.e., more relaxed distortion constraints) imply that stronger compression levels (lower $I(T;X)$ values) are attainable. Moreover, $R(D)$ is known to be a non-increasing *convex* function of D ([20], page 349) in the *distortion-compression plane* where the horizontal axis corresponds to D and the vertical axis corresponds to $I(T;X)$. Clearly, the function $R(D)$ separates two regions in this plane. The region above the curve (known as the *rate distortion region* of the source) corresponds to all the *achievable* distortion-compression pairs, while the region below the curve is non-achievable. In other words, let $\{D, I\}$ be such a distortion-compression pair. If this pair is located above the curve, there is a compressed representation T with a compression level $I(T;X) = I$ and an expected distortion which is upper bounded by D. Figure 2.2 illustrates these ideas.

Finding the rate-distortion function requires solving a minimization problem of a convex function over the convex set of all the (normalized) conditional distributions $p(t \mid x)$, satisfying the distortion constraint. This problem can be solved by introducing a Lagrange multiplier, β, and then minimizing the functional

$$\mathcal{F}[p(t \mid x)] = I(T;X) + \beta \,\langle\, d(x,t)\, \rangle_{p(x)p(t|x)} \;, \tag{2.3}$$

under the normalization constraints $\sum_t p(t \mid x) = 1,\ \forall x \in \mathcal{X}$. This formulation has the following well known consequences.

Theorem 2.1.1 : *The solution of the variational problem*

$$\frac{\delta \mathcal{F}}{\delta p(t \mid x)} = 0\;,^{1} \tag{2.4}$$

[1] We henceforth use the notation $\frac{\delta \mathcal{F}}{\delta p(t|x)}$ to emphasize that the analysis presented in this thesis can be extended to handle continuous variables as well. Nonetheless, since for simplicity we concentrate on discrete random variables with a finite number of values, an equivalent possible notation is $\frac{\partial \mathcal{F}}{\partial p(t|x)}$.

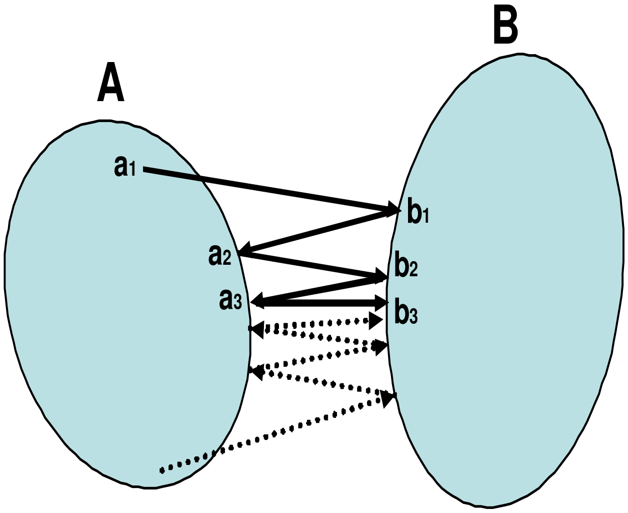

Figure 2.3: An illustration of alternating minimization of the Euclidean distance between two convex sets in $\mathcal{R}^2$. Since the minimized function (i.e., the Euclidean distance between the sets) is convex, the algorithm will always converge to the global minimum distance, independently of the initialization. This is also true for minimizing the KL divergence between two convex sets of probability distributions.

for normalized distributions $p(t \mid x)$ is given by the exponential form

$$p(t \mid x) = \frac{p(t)}{Z(x,\beta)} e^{-\beta d(x,t)} , \tag{2.5}$$

where $Z(x,\beta)$ is a normalization (partition) function. Moreover, the Lagrange multiplier β, determined by the value of D, is positive and satisfies

$$\frac{\delta R}{\delta D} = -\beta . \tag{2.6}$$

Note that this solution is implicit since $p(t)$ on the right hand side of Eq. (2.5), clearly depends on $p(t \mid x)$ through $p(t) = \sum_{x,t} p(x)p(t \mid x)$. The question of how to compute the rate-distortion function is the topic of the next section.

It should be emphasized that the function $R(D)$ is defined with respect to a *fixed* set of representatives, given by T values. In general, choosing a different set of representatives will define a different distortion matrix between X and T values, resulting in a different rate distortion function. The (important) question of how to choose an optimal set of representatives is disregarded in the classical derivative of rate distortion theory.

2.1.2 The Blahut-Arimoto algorithm

Consider the following problem: Given two convex sets A and B in $\mathcal{R}^n$, we would like to find the minimum distance between them. A natural algorithm would be to choose some point $a \in A$ and find the point $b \in B$ that is closest to it, then fix this b and find its closest point in A. Repeating this process must converge since clearly the distance decreases with each iteration (see Figure 2.3). But does it converge to the (global) minimum distance? Csiszár and Tusnády [23] have shown that if the two sets are convex the answer is positive (under some requirements from the distance measure). In particular, for two sets of probability distributions and using the KL divergence as the "distance" measure, the algorithm converges to the minimum KL between the two sets, and is known as the Blahut-Arimoto algorithm [3, 14]. [2]

To apply this algorithm to calculating $R(D)$ one must rewrite this function as a minimum of the KL between two (convex) sets of distributions. To do this we need the following simple proposition.

Proposition 2.1.2 : *Let $p(x)p(t \mid x)$ be a given joint distribution. Then the prior distribution $p(t)$ that minimizes $D_{KL}[p(x)p(t \mid x) \| p(x)p(t)]$ is the corresponding marginal distribution, i.e.,*

$$p^*(t) = \sum_x p(x)p(t \mid x) \ . \tag{2.7}$$

Note that at the minimum, $D_{KL}[p(x)p(t \mid x) \| p(x)p(t)]$ is exactly the information, $I(T;X)$ calculated on the basis of the joint distribution $p(x)p(t \mid x)$. Hence, this KL divergence is an upper bound for the compression-information term, and equality holds if and only if $p(t)$ is set to be the marginal distribution of $p(x)p(t \mid x)$. This proposition allows us to rewrite the definition of the rate-distortion function as a double minimization:

$$R(D) = \min_{\{p(t)\}} \ \min_{\{p(t|x):\ \langle\ d(x,t)\ \rangle \leq D\}} D_{KL}[p(x)p(t \mid x) \| p(x)p(t)] \ . \tag{2.8}$$

If A is the set of all joint distributions $p(t,x)$ with marginal $p(x)$ that satisfy the distortion constraint and if B is the set of the product distributions $p(t)p(x)$ with some normalized $p(t)$, we get

$$R(D) = \min_{b \in B} \ \min_{a \in A} D_{KL}[a \| b] \ . \tag{2.9}$$

We now apply the Blahut-Arimoto algorithm. We start by specifying β which determines D, namely the distortion constraint. We define some initial guess for $p(t)$ (i.e., choose a random point in B), and then use Eq. (2.5) to find $p(t \mid x)$ (i.e., a point in A) that minimizes the information subject to the distortion constraint. Given this distribution we use Eq. (2.7) to find a new $p(t)$ that further minimizes the same information (or KL divergence). Repeating this process monotonically reduces the right hand side of Eq. (2.8). Thus, the algorithm converges to a limit, which was shown by Csiszár to be $R(D)$ where the value of D depends on β [21]. More specifically, using Theorem 2.1.1 we see that the algorithm converges to a unique point on the rate-distortion curve in which the slope of the curve equals $-\beta$ [16]. Choosing different β values in principle enables a numerical estimation of the full curve. For $\beta \to \infty$ we focus solely on minimizing the distortion which corresponds to the extreme case of the rate-distortion curve with $\langle\ d(x,t)\ \rangle_{p(x)p(t|x)} \to 0$. On the other hand, for $\beta \to 0$ we are only interested in compression, which corresponds to the other extreme of the curve with $R \to 0$ (see Eq. (2.3) and Figure 2.2). A Pseudo-code of this algorithm is given in Figure 2.4.

Note that the minimization is done independently in the two sets of distributions. That is, although $p(t)$ depends on $p(t \mid x)$, while minimizing with respect to $p(t \mid x)$ we assume that $p(t)$ is fixed. In the next update step we minimize with respect to $p(t)$ (assuming that $p(t \mid x)$ is fixed) through Eq. (2.7). Only after this step, $p(t)$ is set to the proper marginal of $p(x)p(t \mid x)$.

As already mentioned, this algorithm only deals with the optimal partitioning of $\mathcal{X}$ (induced by $p(t \mid x)$) with respect to a fixed set of representatives (T values). Thus, the set of all possible distortions $d(x,t)$, $\forall x \in \mathcal{X}$, $\forall t \in \mathcal{T}$ is pre-defined and fixed during the process, and the algorithm computes the rate-distortion function with respect to this choice of representatives. In practice, it is also highly important to find the optimal representatives, *given* the partition $p(t \mid x)$. This joint optimization, however, in general does not have a unique solution, as explained later on.

[2] The convergence in this case is due to the fact that the KL divergence is a convex function in both of its arguments *simultaneously*, see [20], page 30.

```
Input:
    Source distribution $p(x)$ .
    Trade-off parameter $\beta$ and convergence parameter $\varepsilon$ .
    A set of representative, given by $T$ values.
    Distortion measure $d : \mathcal{X} \times \mathcal{T} \rightarrow \mathcal{R}^{+}$ ,$\forall x \in \mathcal{X},\ \forall\, t \in \mathcal{T}$ .

Output:
    Value of $R(D)$ where its slope equals $-\beta$.

Initialization:
    Initialize $R^{(0)} \leftarrow \infty$ and randomly initialize $p(t)$ .

While True

    • $P^{(m+1)}(t \mid x) \leftarrow \frac{P^{(m)}(t)}{Z^{(m+1)}(x,\beta)} e^{-\beta d(x,t)}$ , $\forall\, t \in \mathcal{T},\ \forall\, x \in \mathcal{X}$ .

    • $P^{(m+1)}(t) \leftarrow \sum_{x} p(x) P^{(m+1)}(t \mid x)$ , $\forall\, t \in \mathcal{T}$ .

    $R^{(m+1)}(D) = D_{KL}[p(x)p^{(m+1)}(t \mid x) \| p(x)p^{(m+1)}(t)]$ .

    If $(R^{(m)}(D) - R^{(m+1)}(D)) \leq \varepsilon$
        Break .
```

Figure 2.4: Pseudo-code of the Blahut-Arimoto algorithm. The input parameter β determines the trade-off between the compression-information and the expected distortion, and in particular the value of D for which the rate-distortion function is calculated. The algorithm converges to the value of a unique point on the rate-distortion curve in which the slope of the curve equals $-\beta$ [16]. Note that, in general, this curve depends on the choice of the representatives (T values) and on the definition of the distortion measure between these representatives and X values.

2.2 Relevance through another variable: the IB principle

The main drawback of the rate distortion approach is that a distortion measure is a part of the problem setup. Clearly, for a given source $p(x)$, different choices of distortion measure will yield different results, and in particular different rate-distortion functions (even for a fixed set of representatives). Therefore, the characterization of the source statistics through $R(D)$ relies on an "outside" definition which often has nothing to do with the source properties. As a result, interpreting $R(D)$ cannot be separated from the (possibly arbitrary) choice of the distortion measure. It is not even clear what insights can be gained that are solely relevant to the input source. Moreover, in many practical applications such as image and speech coding, natural text analysis etc., defining an "appropriate" distortion measure is far from trivial.

The IB principle, suggested by Tishby, Pereira and Bialek [82] introduces an alternative approach, while trying to cope with these difficulties. The motivation comes from the fact that in many cases, in contrast to defining a distortion measure, defining a "target" variable with respect to X is a rather simple question with a natural answer. Let us elaborate about this point since it is fundamental to the rest of this thesis.

Consider a simple example where $p(x)$ describes the prior probabilities of all the different words in a given collection of news articles. These articles might deal with different topics, and furthermore reflect different writing styles corresponding to the different newspapers. [3] Clearly $|\mathcal{X}|$ might be very large and let us assume that we are interested in finding a compressed representation of X, denoted by T. Without further details this is an ill posed task: What *features* of the original variable, X should be preserved by T? How should we choose the distortion measure between T and X values?

To answer these questions one must provide a more precise description of the task. For example, a reasonable definition would be to require a compression of X that in some sense preserves the "information" contained in the collection of articles. This is still vague if we do not specify "information about what" since clearly a need for information is well defined only with respect to some other signal that we would like to learn more about. Hence, continuing our example, we may look for a compressed representation of X that preserves the information about, e.g., the *topics* present in the corpus. It turns out that formalizing this task is rather simple and is done by defining a new random variable, denoted here by Y. [4] In our example, the values of Y will correspond to all the different topics in the collection. This is our "target" variable, the variable that we are interested in, or the *relevant* variable.

Instead of considering only $p(x)$, we now consider the *joint* statistics, $p(x, y)$ (which can be estimated rather easily in this case). Once this joint distribution is given we can complete the formulation of the problem by suggesting to look for a compressed representation of X which maintains the (mutual) information about the relevant variable, Y as high as possible. The interpretation of the obtained results will now be straightforward: T compresses X while trying to preserve the relevant features in X with respect to the different topics in the corpus. In some sense this formulation forces the "user" to define precisely his goals while compressing X. For example, an entirely alternative task would be to compress X while preserving the information about the different *writing styles* present in the collection. In this case, the values of the relevant variable Y would be all the different writing styles, the estimated joint statistics, $p(x, y)$ would be different and obviously so would the results. Nonetheless, the interpretation of these results would still be objective and clear: T now compresses X while trying to preserve the relevant features in X with respect to the different writing styles in the corpus.

We now turn to a more formal description of the above discussion. As in rate distortion, the compactness of the new representation is measured through the compression-information, $I(T; X)$. However, the distortion upper bound constraint is now replaced by a *lower* bound constraint over the *relevant information*, given by $I(T; Y)$. In other words, we wish to minimize $I(T; X)$ while preserving $I(T; Y)$ above some minimal

[3] For simplicity we assume the standard "bag-of-words" model, i.e., that the prior probability of some word is independent of its neighbors and is estimated based on its relative frequency in the corpus.

[4] For simplicity we will assume that Y is discrete as well, although this assumption is not always necessary.

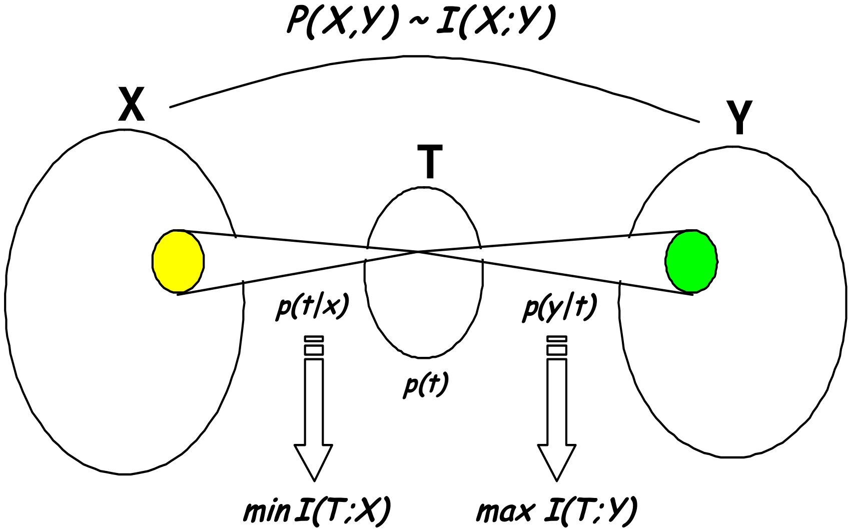

Figure 2.5: The information between X and Y is squeezed through the compact "bottleneck" representation, T. In particular, under some constraint over the minimal level of relevant information, $I(T;Y)$, one is trying to *minimize* the compression-information, $I(T;X)$ (note the similarity of the left part of the figure with Figure 2.1). In this formulation the IB principle extends the rate distortion problem, in the sense that given $p(x,y)$, the setup of the problem is completed and no distortion measure need be defined. An equivalent formulation is to constraint the compression-information to some maximal level, and then try to *maximize* the relevant information term. In this formulation the IB principle is somewhat reminiscent of the channel coding problem. Specifically, in this case one is trying to maximize the information transmitted through a (compact) channel, where the channel properties are governed by the constraint over the compression-information.

level. In this sense, one is trying to squeeze the information X provides about Y through a compact "bottleneck" formed by the compressed representation, T. An equivalent formulation would be to constrain the compression-information to some maximal level, and then try to *maximize* the relevant information term. Either way, the basic trade-off is between minimizing the compression-information while maximizing the relevant-information. An illustration of this idea is given in Figure 2.5.

The first obvious observation is that since T is a compressed representation of X it should be completely defined given X alone. That is, $p(t \mid x,y) = p(t \mid x)$ which implies

$$p(x,y,t) = p(x,y)p(t \mid x) \ . \tag{2.10}$$

An equivalent formulation is to require the following Markovian independence relation, which we will term the IB Markovian relation:

$$T \leftrightarrow X \leftrightarrow Y \ .^{5} \tag{2.11}$$

Obviously the lossy compression T cannot convey more information than what is included in the original data; that is, since T depends only on X it cannot provide any "new" information about Y, except for the information already given by X. Using Data Processing Inequality ([20], page 32) and the above IB Markovian relation it follows $I(T;Y) \leq I(X;Y)$ which formally expresses this intuition.

[5] As noted in [82], it is important to emphasize that this is *not* a modeling assumption about the quantization in T. In fact, this is not an assumption but rather a definition of the problem setup, hence the marginal over $p(x,y,t)$ with respect to X and Y is always consistent with the input distribution, $p(x,y)$. In contrast, the standard modeling approach defines T as a hidden variable in a model of the data, where in this case one *assumes* the Markov independence relation $X \leftrightarrow T \leftrightarrow Y$, which is typically not consistent with the input data. See Section A.5 for further discussion.

Note that in particular the IB Markovian relation characterizes $p(t)$ and $p(y \mid t)$ through

$$\begin{cases} p(t) = \sum_{x,y} p(x,y,t) = \sum_x p(x)p(t \mid x) \\ p(y \mid t) = \frac{1}{p(t)} \sum_x p(x,y,t) = \frac{1}{p(t)} \sum_x p(x,y)p(t \mid x) \ . \end{cases} \tag{2.12}$$

As in rate distortion, we wish to capture the above trade-off in a single variational principle and find the optimal partitioning using the method of Lagrange multipliers. Specifically, Tishby *et al.* [82] suggested the following *IB variational principle*, which we will also term the *IB-functional*:

$$\mathcal{L}[p(t \mid x)] = I(T;X) - \beta I(T;Y) \ , \tag{2.13}$$

where $I(T;X)$, $I(T;Y)$ are defined through $p(t \mid x)$ and Eqs. (2.12). As in rate distortion, β is a Lagrange multiplier controlling the trade-off and the free parameters correspond to the stochastic mapping $p(t \mid x)$. As $\beta \to 0$ we are interested solely in compression, hence all T values collapse to a single value to which all X values are assigned. Clearly, in this case the compression is optimal, $I(T;X) = 0$, but all the relevant information is lost as well, $I(T;Y) = 0$. On the other extreme, as $\beta \to \infty$ we are focused only on preservation of relevant information. In this case the (trivial) solution is where T copies X and through it we obtain $I(T;Y) = I(X;Y)$ which is the upper bound for this term. However, in this case clearly there is no compression since $I(T;X) = H(X)$ is maximized as well. The interesting cases are of course in between, where for finite values of β we are able to extract rather compressed representations of X, while still maintaining a significant fraction of the original information about Y. In Chapter 4 and in Chapter 5 we present several such examples. More generally speaking, by varying the single parameter β one can explore the trade-off between compression and preservation of relevant information for different resolutions. This (monotonic) trade-off is fully characterized by a unique function, termed here the *relevance-compression function*. This function, which is a natural extension of the rate-distortion function, is described in the next section.

Note that the above formulation of the principle is inherently asymmetric. Only X is compressed and only Y serves as a relevant variable. This asymmetry suggests calling this principle the *single-sided* IB principle, which is the title of this thesis part. In Part III we deal with a family of extensions to this principle, among them are more symmetric formulations and more.

2.3 The relevance-compression function

Given a joint probability distribution $p(x,y)$ the IB optimization problem can be stated as follows: find T such that $I(T;X)$ is minimized, under the constraint $I(T;Y) \geq \hat{D}$ (where $T \leftrightarrow X \leftrightarrow Y$). Thus, it is natural to define a mathematical function which is analogous to the rate-distortion function.

Definition 2.3.1: The *relevance-compression function* for a given joint distribution $p(x,y)$ is defined as

$$\hat{R}(\hat{D}) \equiv \min_{\{p(t|x):\, I(T;Y) \geq \hat{D}\}} I(T;X) \ , \tag{2.14}$$

where $T \leftrightarrow X \leftrightarrow Y$ and the minimization is over all the normalized conditional distributions, $p(t \mid x)$ for which the constraint is satisfied.

In words, $\hat{R}(\hat{D})$ is the minimal achievable compression-information, for which the relevant information is above $\hat{D}$. Consider the *relevance-compression plane* where the horizontal axis corresponds to $I(T;X)$ and the vertical axis corresponds to $I(T;Y)$. This plane (termed the *information plane* in [82]) is the natural equivalent to the distortion-compression plane in rate distortion, and we will further refer to the trajectory defined by the relevance-compression function in this plane as the *relevance-compression curve*. [6] From Definition 2.3.1 it is clear that $\hat{R}(\hat{D})$ separates between two regions in this plane. The region below the curve, which we may term the *relevance-compression region* of $p(x,y)$, corresponds to all the *achievable* relevance-compression pairs. In contrast, the region above the curve is non-achievable. In other words, let $\{I_x, I_y\}$ denote some levels of compression-information and relevant information, respectively. If this pair is located below the curve, then (for the given $p(x,y)$) there is some compressed representation T with a compression level $I(T;X) = I_x$ and a relevant information $I(T;Y) = I_y$.

The following proposition shows that the basic properties of the relevance-compression function are similar to those of the rate-distortion function.

Proposition 2.3.2 : *Let $p(x,y)$ be some joint probability distribution. The corresponding relevance-compression function, $\hat{R}(\hat{D})$ is a non-decreasing concave function of $\hat{D}$. Moreover, the slope of this function is determined through*

$$\frac{\delta \hat{D}}{\delta \hat{R}} = \beta^{-1} \ . \tag{2.15}$$

As a result, the slope of the curve corresponding to $\hat{R}(\hat{D})$ gradually decreases while we shift our preferences from compression to preservation of relevant information. Starting at the maximal compression end, $\beta \to 0$ and the slope approaches ∞. At the other end, all the relevant information is preserved, $\beta \to \infty$ and the slope of the curve approaches 0.

Alternatively we may consider the cardinality of the compression variable, $|\mathcal{T}|$ which increases monotonically along the curve. At the maximal compression end we look for the most compact representation, i.e., $|\mathcal{T}| = 1$. By gradually increasing β the constraint over $I(T;Y)$ becomes more demanding. At some finite (critical) β value, this constraint guides the system to focus not only on compression but also on the relevant information term. Consequently, the single value of T *bifurcates* into two separate value, that fulfill the relevant information constraint. This phenomenon is a phase-transition of the system. Successive increases of β will reach additional phase transitions in which additional splits of some values of T emerge. At the limit $\beta \to \infty$, the system concentrates only on the relevant information term, T simply copies X and its cardinality reaches its maximal required level, $|\mathcal{T}| = |\mathcal{X}|$.

One immediate outcome of this discussion is that in principle one can define a family of sub-optimal characteristic curves, where each one corresponds to an additional constraint over the cardinality $|\mathcal{T}|$. For example, constraining this value to be upper bounded by 2 will yield a curve that is originally identical with $\hat{R}(\hat{D})$. At the critical value of β for which the two values of T split into three values, this curve separates from $\hat{R}(\hat{D})$ and continues as a sub-optimal trajectory in the relevance-compression plane. The limit value of this curve as $\beta \to \infty$ reflects the most informative solution that can be found with only two values (or two clusters) in T. Moreover, as discussed in the next section, this solution is deterministic, meaning that each value of X is assigned to one value of T with probability 1, and to the second value with probability 0. An illustration of the above discussion is given in the left panel of Figure 2.6.

As mentioned in the previous section, $I(T;Y)$ is always upper bounded by the original information, $I(X;Y)$. Additionally, $I(T;X)$ is clearly upper bounded by the original compression-information, $I(X;X) = H(X)$ (see Section 1.2.3). Therefore, in many cases it is also valuable to consider the *normalized* relevance-compression plane (and function), where now the vertical axis is determined by $\frac{I(T;Y)}{I(X;Y)}$ while the horizontal axis corresponds to $\frac{I(T;X)}{H(X)}$. The normalized relevance-compression function is thus always bounded between one and zero, hence different joint distributions $p(x,y)$ can be characterized and compared by their corresponding curves in these normalized plane. Roughly speaking, we may say that the existence of a

[6] In rate distortion the vertical axis corresponds to $I(T;X)$ but Tishby *et al.* [82] defined the *horizontal* axis to measure this quantity. For consistency with their work we maintain this convention. As a result, the characteristic curves discussed in this section are concave, with a monotonically *decreasing* positive slope, whereas the standard presentation of the rate-distortion function is as a convex curve with a monotonically *increasing* negative slope.

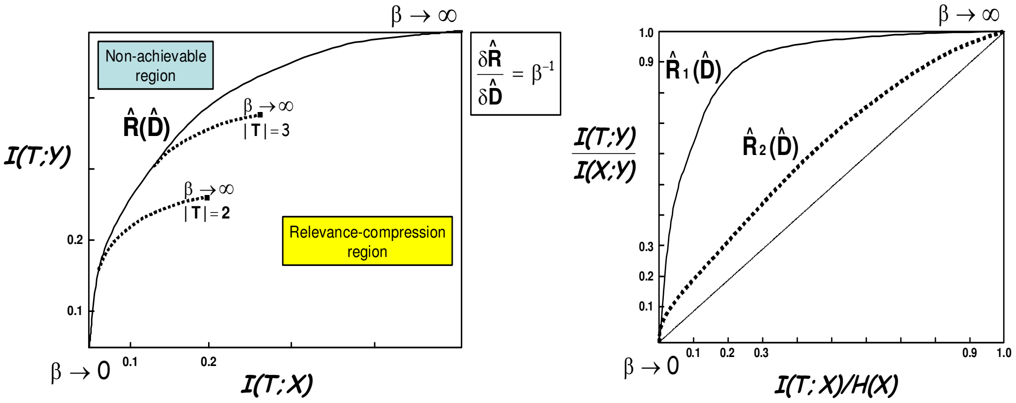

Figure 2.6: **Left:** An illustration of a relevance-compression function, $\hat{R}(\hat{D})$. This function defines a monotonic concave curve in the relevance-compression plane (solid line in the figure). The region below the curve is achievable while the region above it is non-achievable. An additional constraint over the number of clusters, $|\mathcal{T}|$, defines additional sub-optimal curves in this plane (dotted lines in the figure). These curves fully characterize the input (source) statistics, $p(x,y)$ in terms of compression versus preservation of relevance information. **Right:** Different joint distributions will generally yield different curves in the *normalized* relevance-compression plane. A "natural structure" in $p(x,y)$ means that most of the relevant information can be captured by a relatively compact representation. This in turn yields a characteristic curve in the form of $\hat{R}_1(\hat{D})$. On the other hand, if any attempt to compress X loses a significant fraction of the relevant information about Y, the corresponding curve ($\hat{R}_2(\hat{D})$ in the figure) will be near the main diagonal of the normalized plane.

"natural structure" in $p(x,y)$ means that most of the relevant information can be captured by a relatively compact representation. This in turn yields a normalized relevance-compression curve which is near unity even when $\frac{I(T;X)}{H(X)}$ is small. On the opposite extreme, if any attempt to compress X loses a significant fraction of the relevant information about Y, the corresponding curve will be near the main diagonal of this plane. In the right panel of Figure 2.6 we illustrate these two cases. Note that analyzing joint distributions in the normalized relevance-compression plane must always be accompanied by considering the absolute information values.

Finally, we should emphasize the basic distinction between the relevance-compression and the rate-distortion functions. In contrast to rate distortion, the relevance-compression characteristic function is based purely on the "input" statistics. Indeed, the assumption about the input is somewhat more challenging, since we assume we have access to the joint distribution $p(x,y)$, not only to $p(x)$. Nonetheless, once this distribution (or a reasonable estimate of it) is available, the problem setup is completed. No distortion measure or any other "outside" (not statistically oriented) definitions are required and $\hat{R}(\hat{D})$ with its sub-optimal variants fully characterizes $p(x,y)$ in terms of compression versus preservation of relevant information.

2.4 Characterizing the solution to the IB principle

In Section 2.1.1 we saw that it is possible to characterize the general form of the optimal solution to the rate distortion problem (Theorem 2.1.1). Is it possible to describe an analogous result to the IB problem? An immediate obstacle is the fact that in rate distortion the problem setup includes the definition of a distortion measure which is also present in the form of the optimal solution. On the other hand, in the IB case no distortion measure is provided in advance. Moreover, while the constraint in rate distortion is linear in the desired mapping, $p(t \mid x)$ (see Eq. (2.1)), this is not the case for the IB problem. Specifically, the dependency of $I(T;Y)$ in $p(t \mid x)$ is *non-linear*, and as a result the corresponding variational problem is in principle much harder.

In spite of these potential pitfalls, Tishby *et al.* [82] introduced a complete formal characterization of the optimal solution to the IB problem which is given in the following theorem.

Theorem 2.4.1: *Assume that $p(x, y)$ and β are given and that $T \leftrightarrow X \leftrightarrow Y$. The conditional distribution $p(t \mid x)$ is a stationary point of $\mathcal{L} = I(T; X) - \beta I(T; Y)$ if and only if*

$$p(t \mid x) = \frac{p(t)}{Z(x, \beta)} e^{-\beta D_{KL}[p(y|x) \| p(y|t)]}, \forall t \in \mathcal{T}, \; \forall x \in \mathcal{X} \; , \tag{2.16}$$

where as before, $Z(x, \beta)$ is a normalization (partition) function.

Clearly this is a formal solution since $p(t)$, $p(y \mid t)$ are determined implicitly through $p(t \mid x)$ by Eqs. (2.12). Note that these two equations together with Eq. (2.16) determine self-consistently the optimal solution. In particular, the optimization here is also over the cluster representatives, $p(y \mid t)$. This is in contrast to rate distortion theory, where the selection of the representatives is a separate problem. An iterative algorithm that constructs a (locally) optimal solution by iterating over these three sets of equations is described in Section 3.1.

It is important to emphasize that the KL divergence, $D_{KL}[p(y \mid x) \| p(y \mid t)]$ *emerges* as the effective distortion measure from the IB variational principle, rather then being assumed in advance. Therefore, in this sense, this is the correct distortion measure to this problem. The essence of the above theorem is that it defines $p(t \mid x)$ in terms of this measure. When $p(y \mid t)$ becomes more similar to $p(y \mid x)$ we may say that the performance of t as a representative of x is improved. In this case the corresponding KL decreases and consequently the membership probability $p(t \mid x)$ increases. On the other hand, if t is not a good representative of x the corresponding KL is large and the membership probability $p(t \mid x)$ is reduced accordingly.

Considering again Eq. (2.16) we see that the value of β determines how diffused the conditional distributions $p(t \mid x)$ are. Small values of β imply high diffusion since β reduces the differences between the KL distortions for different values of T. In the limit $\beta \to 0$ there is maximal diffusion and in fact $p(t \mid x)$ does not depend on the value of X. This effectively means a single value in $\mathcal{T}$, i.e., maximal compression.

As β increases most of the conditional probability mass is assigned to the value t with the smallest KL distortion. In the limit $\beta \to \infty$ this value will contain all the probability mass, i.e., $p(t \mid x)$ becomes deterministic and every value of X is assigned to a single value of T with a probability of one. This limit corresponds to the extreme end of the relevance-compression curves where all the emphasis is on preservation of relevant information.

As already mentioned, the characterization of the optimal solution is a formal characterization. The question of how to construct optimal or approximated solutions to the IB problem in practice, is the topic of the next chapter.

Chapter 3

IB Algorithms

We now consider algorithms for constructing exact or approximated solutions to the IB variational principle. We describe four different complementary approaches to this task. The first two were originally suggested in [60, 82] and are presented in detail in the first two sections of this chapter. The other two approaches are novel, and were first introduced in [74, 76]. These two methods are described in Section 3.3 and Section 3.4, respectively. In Section 3.5 we discuss the relationships between the different approaches, and in the last section of this chapter we show that combinations of these approaches are also plausible in some cases.

3.1 iIB: an iterative optimization algorithm

We start with the case where β is fixed. In this case, following standard strategy in variational methods, we simply apply the fixed-point equations given in Eq.(2.16). More precisely, we use an iterative algorithm, that at the m'th iteration maintains the conditional distributions $\{P^{(m)}(t \mid x)\}$. At the $m+1$'th iteration, the algorithm applies an update step:

$$P^{(m+1)}(t \mid x) \leftarrow \frac{P^{(m)}(t)}{Z^{(m+1)}(x,\beta)} e^{-\beta D_{KL}[p(y|x) \| P^{(m)}(y|t)]} \tag{3.1}$$

where $P^{(m)}(t)$ and $P^{(m)}(y \mid t)$ are computed using the conditional probabilities $\{P^{(m)}(t \mid x)\}$, and the IB Markovian relation, $T \leftrightarrow X \leftrightarrow Y$. Specifically, following Eqs. (2.12) we have

$$\begin{cases} P^{(m)}(t) = \sum_x p(x) P^{(m)}(t \mid x) \\ \\ P^{(m)}(y \mid t) = \frac{1}{P^{(m)}(t)} \sum_x P^{(m)}(t \mid x) p(x,y) \ . \end{cases} \tag{3.2}$$

We will term this algorithm the iterative IB (iIB) algorithm. A Pseudo-code is given in Figure 3.1 and an illustration of the process is given in Figure 3.2.

Note that this algorithm is a natural extension of the Blahut-Arimoto algorithm (Section 2.1.2). However, there is an important distinction. The Blahut-Arimoto algorithm, in principle allows to converge to a point on the relevance-compression curve in which the slope is $-\beta$. This curve is defined with respect to a given distortion measure and a given fixed set of representatives (T values). Hence, the alternating minimization is done only over the sets $\{p(t \mid x)\}$, $\{p(t)\}$. In contrast, the iIB algorithm tries to converge to a point on the relevance-compression curve in which the slope equals β^{-1}. This curve does *not* depend on a pre-definition of a distortion measure, non on fixing the set of representatives. In particular, in the iIB algorithm the minimization is additionally over the set of representative distributions (or clusters centroids), $\{p(y \mid t)\}$. As a result, in general there is no guarantee of the uniqueness of the solution, and all one can expect is to converge to a *locally* optimal solution, as explained in the next section.

```
Input:
    Joint distribution $p(x, y)$ .
    Trade-off parameter $\beta$ .
    Cardinality parameter $M$ and a convergence parameter $\varepsilon$ .

Output:
    A (typically "soft") partition $T$ of $\mathcal{X}$ into $M$ clusters.

Initialization:
    Randomly initialize $p(t \mid x)$ and find the corresponding
    $p(t),\ p(y \mid t)$ through Eqs. (3.2).

While True
    • $P^{(m+1)}(t \mid x) \leftarrow \frac{P^{(m)}(t)}{Z^{(m+1)}(x,\beta)} e^{-\beta D_{KL}[p(y|x) \| p(y|t)]}$ , $\forall\, t \in \mathcal{T},\ \forall\, x \in \mathcal{X}$ .
    • $P^{(m+1)}(t) \leftarrow \sum_x p(x) P^{(m+1)}(t \mid x)$ , $\forall\, t \in \mathcal{T}$ .
    • $P^{(m+1)}(y \mid t) = \frac{1}{P^{(m+1)}(t)} \sum_x P^{(m+1)}(t \mid x) p(x, y)$ , $\forall\, t \in \mathcal{T},\ \forall\, y \in \mathcal{Y}$ .

    If $\forall\, x \in \mathcal{X},\ JS_{\frac{1}{2},\frac{1}{2}}[P^{(m+1)}(t \mid x), P^{(m)}(t \mid x)] \le \varepsilon$ ,
        Break .
```

Figure 3.1: Pseudo-code of the iterative IB (iIB) algorithm. JS denotes the Jensen-Shannon divergence (Definition 1.2.17). In principle we repeat this procedure for different initializations and choose the solution which minimizes $\mathcal{L} = I(T; X) - \beta I(T; Y)$.

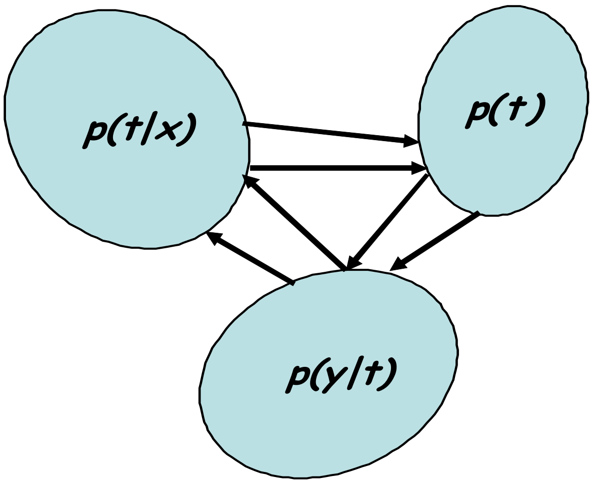

Figure 3.2: Illustration of the iIB algorithm. The (alternating) minimization is performed over three convex sets of distributions. At each step two distributions in two sets are kept constant, and the algorithm finds a third distribution in the third set that further minimizes the IB-functional. Since the IB-functional is *not* convex in the product space of these three sets, different initializations might lead to different local optima.

3.1.1 Convergence of the iIB algorithm

Theorem 3.1.1 : *Iterating over the fixed-point equations given in Eq. (3.1) converges to a stationary fixed point of the IB-functional.*

Proof: Since the proof of this theorem provides further insights about the method, we outline it below (a shorter version already appeared in [82]). We start by introducing the following auxiliary functional:

$$\mathcal{F}_{IB} \equiv - \langle \log Z(x,\beta) \rangle_{p(x)p(t|x)} = -\sum_{x,t} p(x)p(t \mid x) \log Z(x,\beta) \ , \qquad (3.3)$$

where, as before, $Z(x,\beta)$ is the normalization (partition) function of $p(t \mid x)$. In other words, $\mathcal{F}_{IB}$ is (minus) the expectation over the log of the partition functions (and is known in physics as the "free energy" of the system).

The general idea of the proof is to show that updates defined by the iIB algorithm can only reduce $\mathcal{F}_{IB}$, and since $\mathcal{F}_{IB}$ is shown to be lower-bounded, we are guaranteed to converge to a self-consistent solution.

Lemma 3.1.2 : *$\mathcal{F}_{IB}$ is non-negative and convex with respect to each of its arguments independently.*

Proof: Using Eq.(2.16) we find that

$$-\log Z(x,\beta) = \log \frac{p(t \mid x)}{p(t)} + \beta D_{KL}[p(y \mid x) \| p(y \mid t)] \ , \qquad (3.4)$$

Thus we obtain

$$\mathcal{F}_{IB} = \sum_{x,t} p(x)p(t \mid x) \log \frac{p(t \mid x)}{p(t)} + \beta \sum_{x,t} p(x)p(t \mid x) D_{KL}[p(y \mid x) \| p(y \mid t)] \ . \qquad (3.5)$$

Therefore, $\mathcal{F}_{IB}$ is a sum of KL divergences, and in particular non negative. Moreover, by Log sum inequality ([20], page 29) it is easy to verify that the KL divergence is (strictly) convex with respect to each of its arguments. Since a sum of convex functions is also convex, and since $\beta > 0$, we achieve the desired result. ▮

Recall that after updating $p(t)$ by the iIB algorithm, $p(t)$ becomes exactly the marginal of the joint distribution $p(x)p(t \mid x)$. Hence, after this update the first term in $\mathcal{F}_{IB}$ corresponds to $I(T;X)$ (and at any stage it is an upper bound of this information, see Proposition 2.1.2). The second term in $\mathcal{F}_{IB}$ can be considered to be an expected *"relevant-distortion"* term, analogous to the standard expected distortion term in rate distortion (see Eq. (2.1) and Eq. (2.3)). The analogy to rate distortion is now even more evident. However, we should keep in mind that the "relevant-distortion" term is *non-linear* in $p(t \mid x)$ since $p(y \mid t)$ is set through this mapping. As a result, $\mathcal{F}_{IB}$ is *not* convex in all of its three arguments simultaneously. Therefore, in general there might be multiple local optima to this functional (for a given β). It is also straightforward to relate the relevant-distortion term to the relevant information term. Specifically,

$$\sum_{x,t} p(x)p(t \mid x) D_{KL}[p(y \mid x) \| p(y \mid t)] = I(X;Y) - I(T;Y) \ .^{1} \qquad (3.6)$$

Since $I(X;Y)$ is a constant, after updating $p(t)$ and $p(y \mid t)$ by the iIB algorithm we have $\mathcal{F}_{IB} = I(T;X) + \beta(I(X;Y) - I(T;Y)) \propto \mathcal{L}$.

[1] Proof: $\sum_{x,t} p(x)p(t|x)D_{KL}[p(y|x)\|p(y|t)] + I(T;Y) = \sum_{x,y,t} p(x,t)p(y|x) \log \frac{p(y|x)}{p(y|t)} + \sum_{x,y,t} p(x,y,t) \log \frac{p(y|t)}{p(y)} = \sum_{x,y,t} p(x,y,t)(\log \frac{p(y|x)}{p(y|t)} + \log \frac{p(y|t)}{p(y)}) = I(X;Y)$, where in the second step we used the IB Markovian relation.

Lemma 3.1.3: *If any of the iIB update equations changes the corresponding distribution, $\mathcal{F}_{IB}$ is reduced.*

Proof: The iIB update equations are given by Eq. (3.1) and Eqs. (3.2). It is straightforward to verify that the derivatives of $\mathcal{F}_{IB}$ with respect to each of its arguments (under proper normalization constraints), provide exactly these three equations. For example, consider $\tilde{\mathcal{F}}_{IB} \equiv \mathcal{F}_{IB} + \sum_x \lambda(x)[\sum_t p(t \mid x) - 1]$, where the second term corresponds to the normalization constraints. Taking the derivative of $\tilde{\mathcal{F}}_{IB}$ with respect to $p(t \mid x)$ and equating to zero will give exactly Eq. (3.1). A similar procedure for the other arguments of $\mathcal{F}_{IB}$, will yield exactly the other two iIB update equations. We now note that updating by equating some derivative of $\mathcal{F}_{IB}$ to zero (while the other two arguments of $\mathcal{F}_{IB}$ remain constant), can only reduce $\mathcal{F}_{IB}$. This is simply due to the fact that $\mathcal{F}_{IB}$ is strictly convex in each of its arguments (independently) and all its arguments correspond to convex sets. Hence, equating some derivative to zero is equivalent to finding the projection of $\mathcal{F}_{IB}$ in the corresponding convex set. This can only reduce $\mathcal{F}_{IB}$, or leave it unchanged, where in this case the update step has no effect. ▮

Combining the above two lemmas we see that through the iIB updates, $\mathcal{F}_{IB}$ converges to a (local) minimum. At this point all the update steps (including Eq. (3.1)) reach a self-consistent solution. Therefore, from Theorem 2.4.1 we are at a fixed-point of the IB-functional, as required. ▮

A key question is how to initialize the iIB procedure. As already mentioned, different initializations can lead to different solutions which correspond to different *local* stationary points of $\mathcal{L}$. Additionally, in some cases we are interested in exploring a hierarchy of solutions for different values of the trade-off parameter β. Tishby *et al.* [82] suggested addressing these two issues through a deterministic annealing-like procedure [63] which is described in the next section.

3.2 dIB: a deterministic annealing-like algorithm

In general, a deterministic annealing procedure works by iteratively increasing the parameter β and then adapting the solution for the previous value of β to the new one [63]. In our context, this allows the algorithm to "track" the changes in the solution as the system shifts its preferences from compression to preservation of relevant information. [2] In other words, by gradually increasing β the algorithm tries to reconstruct the optimal relevance-compression curve, $\hat{R}(\hat{D})$.

Recall that when $\beta \to 0$, the solution consists of essentially one cluster, i.e., $|\mathcal{T}| = 1$. Successive increases of β will reach consecutive phase transitions in which the current values of T split in order to support the required minimal level of $I(T;Y)$. The general idea is to try to identify these value (i.e., cluster) bifurcations. At the end of the procedure we record the obtained bifurcating structure that traces the sequence of solutions at different values of β (see, for example, Figure 4.3).

The main technical problem is how to detect such bifurcations. One option is at each step to take the solution from the previous step (i.e., for the previous value of β we considered) and construct an initial problem in which we *duplicate* each value of T. To define such an initial setting we need to specify the conditional probabilities of these duplicated values given each value of X. Suppose that t_1 and t_2 are two such duplications of the value t. Then we set $p^*(t_1 \mid x) = p(t \mid x)\left(\frac{1}{2} + \alpha\hat{\epsilon}(t,x)\right)$ and $p^*(t_2 \mid x) = p(t \mid x)\left(\frac{1}{2} - \alpha\hat{\epsilon}(t,x)\right)$, where $\hat{\epsilon}(t,x)$ is a (stochastic) noise term randomly drawn out of $U[-\frac{1}{2},\frac{1}{2}]$ and $\alpha > 0$ is a (typically small) scale parameter. Thus, each copy t_1 and t_2 is a slightly perturbed version of t. If β is high enough, this random perturbation suffices to allow the two copies of t to diverge. If β is too small to support such bifurcation, both perturbed versions will collapse to the same solution.

After constructing this initial point, we iteratively perform the update equations of the iIB algorithm until convergence. If after the convergence the behavior of t_1 and t_2 is sufficiently different then we declare that

[2] In deterministic annealing terminology, $\frac{1}{\beta}$ is the "temperature" of the system, and thus increasing β corresponds to "cooling" the system.

```
Input:
    Similar to the iIB algorithm.
    Additional Parameters: $\alpha,\ \varepsilon_\beta$, and $d_{min}$

Output:
    (Typically "soft") partitions $T$ of $\mathcal{X}$ into $m = 1, \ldots, M$ clusters.

Initialization:
    $\beta \leftarrow 0$
    $\mathcal{T} \leftarrow \{t\},\ p(t \mid x) = 1$ .

Main annealing loop:
    $\beta \leftarrow f(\beta, \varepsilon_\beta)$

    Duplicate clusters:
    For every $t \in \mathcal{T}$ and every $x \in \mathcal{X}$ ,
        Randomly draw $\hat{\epsilon}(t,x) \sim U[-\frac{1}{2}, \frac{1}{2}]$ and define:
        $p^*(t_1 \mid x) = p(t \mid x)\left(\frac{1}{2} + \alpha\hat{\epsilon}(t,x)\right)$
        $p^*(t_2 \mid x) = p(t \mid x)\left(\frac{1}{2} - \alpha\hat{\epsilon}(t,x)\right)$

    Apply iIB using the *duplicated* cluster set as initialization.

    Check for Splits:
    $\forall\, t \in \mathcal{T}$ , if $JS_{\frac{1}{2},\frac{1}{2}}[p(y \mid t_1), p(y \mid t_2)] \geq d_{min}$ ,
        $\mathcal{T} \leftarrow \{\mathcal{T} \setminus \{t\}\} \cup \{t_1, t_2\}$

    If $|\mathcal{T}| \geq M$ , return.
```

Figure 3.3: Pseudo-code of the deterministic annealing-like algorithm (dIB). JS denotes the Jensen-Shannon divergence (Definition 1.2.17). $f(\beta, \varepsilon_\beta)$ is a simple function used to increment β based on its current value and on some scaling parameter ε_β . d_{min} is a scalar parameter used to determine a bifurcation of two copies of some value into two independent values. Note that in principle this parameter should be set as a function of β and not with a fixed value. The algorithm stops when the maximal cardinality of T is exceeded. Alternatively, it is possible to use generalization considerations to limit the maximal value of β (see the discussion in Section 6.2.2).

the value t has split, and incorporate t_1 and t_2 into the bifurcation hierarchy we construct for T. Finally, we increase β and repeat the whole process. We will term this algorithm the *dIB* algorithm. A Pseudo-code is given in Figure 3.3.

There are several technical difficulties with applying this algorithm. First, several parameters (see Figure 3.3) must be tuned to detect cluster splits. Setting these parameters without any prior knowledge about the data is not a trivial issue. Moreover, it is not a priory clear that these parameters should be fixed during the process. A possible alternative is to set them as a function of β (see Section 4.3 for an example). Additionally, the rate of increasing β should be tuned, otherwise cluster splits might be "skipped" by the process. Lastly, the duplication process is stochastic in nature (and involves additional parameters) which in principle is not a desirable property of a clustering procedure.

In the following section we describe a much simpler procedure. This is a fully deterministic non-parametric approach. However, in contrast to the dIB algorithm, the extracted solutions have no guarantee of being even a local stationary point of the IB-functional. Thus, we oppose an approximated simple algorithm, versus an exact complicated one.

3.3 aIB: an agglomerative algorithm

The agglomerative Information Bottleneck (aIB) algorithm employs a greedy agglomerative clustering technique to find a hierarchical clustering tree in a *bottom-up* fashion. In several works it has been shown to be useful for a variety of real-world problems, including supervised and unsupervised text classification [77, 78, 87], gene expression analysis [83], neural code analysis [67, 68], image clustering [38], protein sequence analysis [56], natural language processing [41] and galaxy spectra analysis [75]. In this section, following the preliminary work in [76], we present this algorithm in detail. For consistency with [76] we consider the problem of *maximizing*

$$\mathcal{L}_{max} = I(T;Y) - \beta^{-1} I(T;X) \; , \tag{3.7}$$

which is clearly equivalent for minimizing the IB-functional defined by Eq. (2.13) (dividing Eq. (2.13) by $-\beta$ yields Eq. (3.7)).

We consider a procedure that typically starts with the most fine-grained solution where $T = X$. That is, each value of X is assigned to a unique singleton cluster in T. Following this initialization we iteratively reduce the cardinality of T by *merging* two values t_i and t_j into a single value $\bar{t}$. To formalize this notion we need to specify the membership probability of the new cluster resulting from the merger $\{t_i, t_j\} \Rightarrow \bar{t}$. This is done rather naturally through

$$p(\bar{t} \mid x) = p(t_i \mid x) + p(t_j \mid x) \; , \forall x \in \mathcal{X} \; . \tag{3.8}$$

In other words, we view the event $\bar{t}$ as the union of the events t_i and t_j.

Using this specification and the IB Markovian relation we can characterize the prior probability and the centroid distribution of the new cluster. This is done through the following simple proposition.

Proposition 3.3.1: *Let $\{t_i, t_j\} \Rightarrow \bar{t}$ be some merger in T defined through Eq. (3.8). If $T \leftrightarrow X \leftrightarrow Y$ then*

$$p(\bar{t}) = p(t_i) + p(t_j) \; , \tag{3.9}$$

and

$$p(y \mid \bar{t}) = \pi_i \cdot p(y \mid t_i) + \pi_j \cdot p(y \mid t_j) \; , \tag{3.10}$$

where

$$\Pi = \{\pi_i, \, \pi_j\} = \{\frac{p(t_i)}{p(\bar{t})}, \frac{p(t_j)}{p(\bar{t})}\} \; , \tag{3.11}$$

is the "merger distribution".

In particular, this proposition together with Eq. (3.8) allows us to calculate $I(T;X), \; I(T;Y)$ after each merger. Also note that the merger distribution, denoted by Π is indeed a proper normalized distribution.

The basic question in an agglomerative process is of course which pair to merge at each step. The merger "cost" in our terms is exactly the difference between the values of $\mathcal{L}_{max}$, before and after the merger. Let T^{bef} and T^{aft} denote the random variables that correspond to T, before and after the merger, respectively. Thus, the corresponding values of $\mathcal{L}_{max}$ are calculated based on T^{bef} and T^{aft}. The merger cost is then defined by,

$$\Delta\mathcal{L}_{max}(t_i, t_j) = \mathcal{L}_{max}^{bef} - \mathcal{L}_{max}^{aft} \; . \tag{3.12}$$

The greedy procedure evaluates all the potential mergers in T and then applies the best one (i.e., the one that minimizes $\Delta\mathcal{L}_{max}(t_i, t_j)$). This is repeated until T degenerates into a single value. The resulting tree describes a range of clustering solutions at all the different resolutions.

```
Input:
    Joint distribution $p(x, y)$ .
    Trade-off parameter $\beta$.

Output:
    Partitions $T$ of $\mathcal{X}$ into $m = 1, \ldots, |\mathcal{X}|$ clusters.

Initialization:
    $T \leftarrow X$ .
    $\forall\, t_i, t_j \in T$ calculate $\Delta\mathcal{L}_{max}(t_i, t_j) = p(\bar{t}) \cdot \bar{d}(t_i, t_j)$ .

Main Loop:
    While $|\mathcal{T}| > 1$
        $\{i, j\} = argmin_{i',j'} \Delta\mathcal{L}_{max}(t_{i'}, t_{j'})$ .
        Merge $\{t_i, t_j\} \Rightarrow \bar{t}$ in $T$ .
        Calculate $\Delta\mathcal{L}_{max}(\bar{t}, t),\ \forall t \in \mathcal{T}$ .
```

Figure 3.4: Pseudo-code of the agglomerative IB (aIB) algorithm.

3.3.1 A local merging criterion

In the procedure outlined above, at every step there are $O(|\mathcal{T}|^2)$ possible mergers of values of T. Since at the initialization, $|\mathcal{T}| = |\mathcal{X}|$, a direct calculation (through Eq. (3.12)) of all the potential merger costs might be unfeasible if $|\mathcal{X}|$ is relatively large.

However, it turns out that one may calculate $\Delta\mathcal{L}_{max}(t_i, t_j)$ while examining only the probability distributions that involve t_i and t_j directly.

Proposition 3.3.2: *Let $t_i, t_j \in T$ be two clusters. Then,*

$$\Delta\mathcal{L}_{max}(t_i, t_j) = p(\bar{t}) \cdot \bar{d}(t_i, t_j) \;, \tag{3.13}$$

where

$$\bar{d}(t_i, t_j) \equiv JS_{\Pi}[p(y \mid t_i), p(y \mid t_j)] - \beta^{-1} JS_{\Pi}[p(x \mid t_i), p(x \mid t_j)] \;. \tag{3.14}$$

Thus, the merger cost is a multiplication of the "weight" of the merger components, $p(\bar{t})$, with their "distance" given by $\bar{d}(t_i, t_j)$. Due to the properties of the JS divergence this "distance" is symmetric but it is not a metric. In addition, its two components have opposite signs. Thus, the "distance" between two clusters is a trade-off between these two factors. Roughly speaking, we may say that it is minimized for pairs that give similar predictions about the relevance variable Y and have different predictions, or minimum overlap about the compressed variable, X. Note that for $\beta^{-1} \to 0$ we get exactly the algorithm presented in [76]. Also note that after applying a merger we need only calculate the merger costs with respect to the new resulting cluster, $\bar{t}$, while all the other costs remain unchanged. A Pseudo-code of this algorithm is given in Figure 3.4.

An important special case is the "hard" clustering case where T is a *deterministic* function of X. That is, $p(t \mid x)$ can only take values of zero or one, meaning every $x \in \mathcal{X}$ is assigned to exactly one cluster $t \in \mathcal{T}$ with a probability of one and to all the others with a probability of zero. Clearly in this case $H(T \mid X) = 0$, hence $I(T; X) = H(T)$. Namely, we are trying to minimize $H(T)$ while preserving $I(T; Y)$ as high as possible. As already mentioned in Section 1.2.2, $H[p]$ tends to decrease for "less balanced" probability distributions p. Therefore, increasing β^{-1} results in a tendency to look for less balanced "hard"partitions and vice versa. A typical result consists of one big cluster and many additional small clusters. Since the algorithm also aims at maximizing $I(T; Y)$ the big cluster usually consists of the values of X which are

Table 3.1: An example for sub-optimality of aIB. Taking $\beta^{-1} = 0$ and assuming $p(x) = \frac{1}{|\mathcal{X}|}$, aIB will first merge $(x_2, x_3) \Rightarrow \bar{t}$, and then will merge $\bar{t}$ with x_1. This yields two clusters with $I(T;Y) \approx 0.017$ which is 62% of the original relevant information, $I(X;Y)$. However, merging x_1 with x_2, and then x_3 with x_4, yields a partition T such that $I(T;Y) \approx 0.021$, which is 77% of the original information. In other words, the greedy aIB procedure is tempted by the "best" merger in the first step, leading to a rather poor merger in the second step, and to an overall sub-optimal result.

.

X	$p(y \mid x)$
x_1	$[\,0.50\ \ 0.50\,]$
x_2	$[\,0.61\ \ 0.39\,]$
x_3	$[\,0.70\ \ 0.30\,]$
x_4	$[\,0.80\ \ 0.20\,]$

less informative about Y. Thus, a value of X must be highly informative about Y to stay out of this cluster. In this sense, increasing β^{-1} is equivalent to inducing a "noise-filter", that leaves only the most relevant features of X in specific clusters. A demonstration of this effect is given in Section 4.2. It is also worth mentioning that in this "hard" clustering case the second term in $\bar{d}(t_i, t_j)$ is simplified through $JS_{\Pi}[p(x \mid t_i), p(x \mid t_j)] = H[\Pi]$.

3.4 sIB: a sequential optimization algorithm

There are two main difficulties in applying an agglomerative approach. First, an agglomerative procedure is greedy in nature, and as such there is no guarantee it will find even a locally optimal solution (see Table 3.1 for an example). In fact, it may not even find a "stable" solution, in the sense that each object belongs to the cluster it is "most similar" to. Second, the time complexity of this procedure is typically on the order of $O(|\mathcal{X}|^3|\mathcal{Y}|)$, and the space complexity is $O(|\mathcal{X}|^2)$, which makes it unfeasible for relatively large datasets. In [74], Slonim *et al.* describe a simple framework for casting a known agglomerative algorithm into a "sequential K-means like" algorithm. In particular, the resulting algorithm is guaranteed (under some loosely restricted conditions) to converge to a "stable" solution in time and space complexity which are significantly better than those of an agglomerative procedure. Following this work we describe here in detail how to apply this idea in our context.

Unlike agglomerative clustering, the sequential procedure maintains a (flat) partition in T with exactly M clusters. The initialization of T can be based upon a random partition of $\mathcal{X}$ into M clusters, or alternatively by employing more sophisticated initialization techniques (e.g., [18]). Additionally, in principle it is possible to use some (non-optimal) output of any other IB algorithm as the initialization.

Given the initial partition, at each step we "draw" some $x \in \mathcal{X}$ from its current cluster $t(x)$ and represent it as a new singleton cluster. [3] Using our known greedy agglomeration procedure (Eq. (3.13)), we can now merge x into t^{new} such that $t^{new} = argmin_{t \in \mathcal{T}} \Delta\mathcal{L}_{max}(\{x\}, t)$, to obtain a (possibly new) partition T^{new}, with the appropriate cardinality. Assuming that $t^{new} \neq t(x)$ it is easy to verify that this step increases the value of the functional $\mathcal{L}_{max}$ defined in Eq. (3.7). Since for any finite β this functional is upper bounded, this sequential procedure is guaranteed to converge to a "stable" solution in the sense that no more assignment updates can further improve $\mathcal{L}_{max}$.

What is the complexity of this approach? In every "draw-and-merge" step we need to calculate the merger costs with respect to each cluster in T, which is on the order of $O(|\mathcal{T}||\mathcal{Y}|)$. Our time complexity is thus

[3] For simplicity we describe this algorithm for the case of "hard" clustering. In principle it is possible to extend this approach to handle "soft" clustering as well.

```
Input:
    Joint distribution $p(x,y)$ .
    Trade-off parameter $\beta$.
    Cardinality value $M$ .

Output:
    A partition $T$ of $\mathcal{X}$ into $M$ clusters.

Initialization:
    $T \leftarrow$ random partition of $\mathcal{X}$ into $M$ clusters.

Main Loop:
    While not $Done$
        $Done \leftarrow TRUE$ .
        For every $x \in \mathcal{X}$ :
                Remove $x$ from current cluster, $t(x)$ .
                $t^{new}(x) = argmin_{t \in \mathcal{T}} \Delta\mathcal{L}_{max}(\{x\}, t)$
                If $t^{new}(x) \neq t(x)$,
                        $Done \leftarrow FALSE$ .
                Merge $x$ into $t^{new}(x)$
```

Figure 3.5: Pseudo-code of the sequential IB (sIB) algorithm. In principle we repeat this procedure for different initializations and choose the solution which maximizes $\mathcal{L}_{max} = I(T;Y) - \beta^{-1} I(T;X)$.

bounded by $O(\ell\ |\mathcal{X}||\mathcal{T}||\mathcal{Y}|)$ where ℓ is the number of loops we should perform over $\mathcal{X}$ until convergence is attained. Since typically $\ell \cdot |\mathcal{T}| \ll |\mathcal{X}|^2$ we get significant run time improvement.

Additionally, we dramatically improve our memory consumption to be on the order of $O(|\mathcal{T}|^2)$.

As in the case of iIB, to reduce the potential sensitivity for local optima, we can repeat this procedure for N different random initializations of T to obtain N different solutions, from which we choose the one which maximizes $\mathcal{L}_{max}$. We will term this algorithm the sequential IB (sIB) algorithm. A Pseudo-code is given in Figure 3.5.

Note that as for the aIB algorithm, a straightforward implementation of this algorithm would result in "hard" clustering solutions. In this case, reducing β will yield less balanced partitions while increasing β will have the opposite effect.

3.5 Relations between the different algorithms

Several relationships between the above mentioned algorithms should be noted specifically. We first note the difference between the merging criterion of the aIB algorithm (Eq. (3.13)), and the effective distortion measure that controls the iIB algorithm, given in Eq. (3.1). In the iIB case the optimization is governed by the KL divergences between data and cluster centroids (or by the likelihood that the data were generated by the centroid distribution). On the other hand, for the aIB algorithm the optimization is controlled through the JS divergences, i.e., the likelihood that the two clusters now being merged have a common source (see Section 1.2.5).

The aIB approach is the simplest and most easy to use method. It is completely non-parametric (except for the need to specify β) and fully deterministic. Moreover, it provides a full clustering *tree* hierarchy. This agglomerative approach is different in several respects from the deterministic annealing-like algorithm. In the dIB case, by "cooling" (i.e., increasing) β, we move along a trade-off curve, from the trivial (single cluster) solution toward solutions with higher resolutions that preserve more relevant information. In contrast, when using aIB we progress in the opposite direction. We start with high resolution clustering and as the merging process proceeds we move toward coarser solutions. During this process β is kept constant and the driving force is the reduction in the cardinality of T. In particular this allows us to look for solutions in different resolutions for a *fixed* trade-off parameter β which is not possible while using dIB. However, in many real-world applications, complexity considerations might rule out using such an agglomerative technique.

In many practical situations the requested number of clusters already implies a significant compression, namely $|\mathcal{T}| \ll |X|$. In these cases, one might be interested in maximizing the relevant information term for the given number of clusters, without inducing a further constraint over the compression information. A simple way to achieve this is to take $\beta \to \infty$ (or $\beta^{-1} \to 0$) while forcing T to the appropriate cardinality. The natural choice in these cases is to use the sIB or the aIB algorithms, that can be easily applied with these β values.

The sIB algorithm is in some sense similar to the iterative optimization algorithm, iIB. Both algorithms provide a stable solution for a fixed cardinality value and a fixed β value. However, there is a clear distinction between these two approaches. The iIB algorithm applies *parallel* update steps. In this scheme, we first update *all* the membership probabilities, $p(t \mid x)$ and only then update the distributions, $p(t)$ and $p(y \mid t)$. On the other hand, sIB applies *sequential* update steps. Under this routine, after every single assignment update, the corresponding centroid and prior distributions are updated as well. In this context the sIB approach seems to have some relations to the *incremental* variant of the EM algorithm for maximum likelihood [54], which still needs to be explored.

Another distinction between both approaches is of course due to the fact that iIB extracts "soft" clustering solutions, while a typical implementation of sIB extracts "hard" solutions. Nonetheless, as mentioned earlier, as β increases, the partitions considered by iIB become (approximately) deterministic. In this case, the analogy between both algorithms is more evident. A natural question is whether locally optimal solutions obtained by one algorithm would be considered as locally optimal by the other. It turns out that the answer to this question is negative. In other words, an (approximately "hard") optimal solution found by the iIB algorithm might be further improved by the sIB algorithm. A concrete example is given in Table 3.2. This example demonstrates the possible presence of multiple local optima to the IB-functional. In particular, if the iIB algorithm is trapped in such a locally optimal solution (due to "unlucky" initializations), it has no way out since it is not concerned with its surroundings. Once an optimal solution is found, no stochastic mechanism is applied to check whether better optima are available. On the other hand, sIB always considers *all* the available local steps from its current solution. In this sense, it always explores the full discrete grid of "hard" solutions surrounding its current solution. If one of these solutions is superior, the sIB will identify it and will perform the necessary update. This property suggests that the sIB will in general be less sensitive to the presence of local optima (even for medium β values) than the iIB algorithm. In Section 4.4 we present some empirical evidence supporting this suggestion.

Although in many applications one may be satisfied with "hard" clustering solutions, it should be emphasized that for any *finite* choice of β, these solutions are typically not the global optimum of the IB-functional. This observation is an immediate corollary of Theorem 2.4.1. Specifically, the mappings $p(t \mid x)$ that correspond to stationary fixed-points of the IB-functional are stochastic in nature (for a finite β). In particular this implies that the global optimum is typically stochastic. Given this fact, it is important to keep in mind that in principle the basic analysis regarding the aIB and the sIB algorithms (including the derivation of the local merging criterion) holds for "soft" clustering as well. This raises the possibility of using an agglomerative procedure over "soft" clustering partitions, which is left for future research. Alternatively we may use "hard" clustering solutions as a platform to extract (optimal) "soft" clustering solutions. This alternative is discussed in the next section.

The above four approaches define an arsenal of heuristics which are all aimed at optimizing the same target functional, the IB-functional. Each of these heuristics has different advantages and disadvantages, and none of them guarantees a *globally* optimal solution (in general this is an NP-hard problem [33]). Generally

Table 3.2: An example demonstrating that a locally optimal solution found by the iIB algorithm might be further improved by the sIB algorithm. The left column indicates the value in X. The next two columns describe the input distribution, given by $p(x)$ and $p(y \mid x)$, respectively. The last two columns describe a locally optimal solution found by the iIB algorithm for $\beta = 50$ and $|\mathcal{T}| = 2$. Since β is relatively high, the solution is approximately deterministic. In particular, every x is assigned with the T value that minimizes the corresponding KL divergence. Nonetheless, this is not an sIB optimal partition. Specifically, performing a draw-and-merge step for x_1 we see that the cost of merging it back to t_1 (for $\beta = 50$) is ≈ 0.004, but the cost of merging it to t_2 is ≈ -0.002. Hence, the sIB algorithm will move x from t_1 to t_2, and by that will indeed improve the relevant information from $I(T;Y) \approx 0.0175$ toward $I(T;Y) \approx 0.028$ (which is about 80% of the original information, $I(X;Y)$). Although the compression-information, $I(T;X)$ is smaller for the iIB solution (0.32 versus 0.69 respectively), if we consider the complete trade-off described by the IB-functional, we see that the iIB solution is clearly inferior. Specifically, for $\beta = 50$ we get $\mathcal{L} = -0.55$ for the iIB solution, and $\mathcal{L} = -0.71$ for the sIB solution.

.

x	$p(x)$	$p(y \mid x)$	$p(t_1 \mid x)$	$p(t_2 \mid x)$
x_1	0.45	$[\,0.4\;\;0.6\,]$	0.998	0.002
x_2	0.45	$[\,0.6\;\;0.4\,]$	1.000	0.000
x_3	0.10	$[\,0.2\;\;0.8\,]$	0.001	0.999

speaking, the question of which algorithm or combination of algorithms to use, given some input joint distribution, depends on the specific data and on the user's goals and resources. It should be stressed that the above approaches are certainly not a complete list, and other algorithms might be employed under the same information theoretic framework. Several such algorithms, aiming at optimizing similar functionals can be found in [6, 25, 30, 35, 58, 81].

3.6 Combining algorithms and reverse-annealing

An obvious observation is that combining different algorithms might be beneficial in certain circumstances. Two such examples are discussed in this section.

As already mentioned, the aIB algorithm is not guaranteed to extract "stable" solutions. However, combining it with the sIB algorithm can overcome this disadvantage. More precisely, we may start by using the aIB algorithm. At some point we can apply sIB, using the current aIB solution as an initial point. This will result in a better solution (in terms of $\mathcal{L}_{max}$) for the same cardinality value. We now may proceed with aIB, later on apply sIB again, and so forth. Roughly speaking, every application of the sIB algorithm "bounces" the solution into the right direction, while correcting previous "bad" moves of the greedy aIB procedure. On the other hand, using the aIB temporary solutions as the input for sIB provides typically good initializations (in contrast to random initializations). In this sense, by this combination, each algorithm attempts to use its relative advantages to compensate for the disadvantages of its companion.

Another possibility for combining different algorithms, first suggested in [76], is extracting optimal "soft" solutions out of "hard" solutions using a *reverse-annealing* process. To do this recall that any stochastic mapping, $p(t \mid x)$ which is characterized through Eq. (2.16), becomes deterministic at the limit $\beta \to \infty$. As a result, given some deterministic mapping $p(t \mid x)$ (found by aIB or sIB), we can use it as a platform to recover a stochastic mapping. Specifically this is done by representing this mapping through Eq. (2.16) with a large enough β (under which the mapping is indeed deterministic for any practical need). We can now use this representation as an initialization to the iIB algorithm, and by this converge to a local stationary point of the IB-functional. We further slightly *reduce* β, use the previous (optimal) $p(t \mid x)$ mapping as an initialization and apply again the iIB algorithm to extract a new mapping which is slightly more stochastic and also corresponds to a stationary point of the IB-functional. Continuing this process we obtain a series of solutions which become more stochastic as β decreases. In principle, these solutions correspond to a curve in the relevance-compression plane, which is upper bounded by the optimal relevance-compression function, $\hat{R}(\hat{D})$ (see Figure 2.6, left panel). In contrast to the curve found by the dIB algorithm, this curve is extracted without the need to identify cluster splits which is rather tricky. Moreover, given some "hard" solution as the initialization, the process of extracting the curve is deterministic (although it might be sensitive to the rate of decreasing β). At the end of this process $\beta \to 0$ and we obtain the most stochastic solution available. In this limit all the cluster representatives are equivalent, i.e., there is effectively just a single value in T and the compression is maximized.

It is important to note the two different roles of β in the above procedure. We first set β to some *fixed* value and use the aIB or the sIB algorithm to find a range of "hard" clustering solutions at different resolutions. This fixed β value controls the value of $H(T)$ in each of these solutions. We now choose *one* of these solutions and use it as an initial point to recover "soft" solutions. To this task we "plug-in" a (new) high value of β into the deterministic mapping $p(t \mid x)$, and by gradually decreasing this parameter, together with applying the iIB algorithm, we extract a series of locally optimal "soft" solutions, which become softer (and more compressed) as β approaches zero. A demonstration of this process is given in Section 4.3.1 (see the right panels of Figure 4.5).

Chapter 4

IB Applications

In this chapter we examine a few applications of the IB algorithms described previously. There are two main purposes to this chapter. First, we would like to examine the different algorithms and the related theoretical concepts put to practical use. Second, we argue that the method is useful for applications with high practical importance, as demonstrated in the last section of this chapter.

An immediate obstacle in applying our theoretical framework is that in practice, typically we do not have access to the true joint distribution $p(x, y)$. Instead, all we have is a finite sample out of this distribution, represented in the form of a count matrix (sometimes termed a contingency table). In the applications presented in this thesis, a pragmatic approach was taken, where we estimated the distribution $p(x, y)$ through a simple normalization of the given count matrix. As we show in the following, our results are satisfactory even in extreme under-sampling situations. Moreover, some of the IB algorithms are well motivated in these situations as well, as we discuss in Section 6.1. Nonetheless, the finite sample effect over our methodology clearly calls for further investigation which is out of the scope of this work.

It is important to keep in mind that an appealing property of the IB framework is that it can be applied to a wide variety of data types in exactly the same manner. There is no need to tailor the algorithms to the specific data, or define a (data specific) distortion measure. Once a reasonable estimate of a joint distribution is provided, the setup is completed. Moreover, the quality of the results can be measured objectively in terms of compression versus preservation of relevant information.

During the research reported in this thesis, many different applications were examined, which are not presented here. These include using word-clusters for supervised and unsupervised text classification [77, 78], galaxy spectra analysis [75], neural code analysis [68], and gene expression data analysis [83]. Additionally, following our preliminary work in [76], new applications have began to emerge in different domains, such as image clustering [38], protein sequence analysis [56], and natural language processing [41, 87]. However, due to the lack of space and for the sake of coherence we concentrate in this chapter on text processing applications that provide a natural testing ground for our methodology.

4.1 sIB for word clustering with respect to different relevant variables

We start with a simple example to demonstrate the effect of choosing different relevant variables on the results. The "4 Universities Data Set" contains $8,282$ WWW-pages collected from computer science departments of various universities by the CMU text learning group. All pages were manually classified into the following topical categories: *'Student', 'Faculty', 'Staff', 'Department', 'Course', 'Project'* and *'Other'* (where we ignored this last "general" topic). However, there is an additional possible classification of these pages, according to the origin universities: *Cornell, Texas, Washington, Wisconsin* or *'Other'* (where again, we ignored the last general category). Therefore, in principle, it is possible to extract (at least) two different word count matrices out of these data. In the first matrix, denoted here as M_{topic}, the counter in each entry indicates the number of occurrences of a specific word in pages that were assigned to one of the six different topics. In the second matrix, denoted here as M_{univ}, the counter in each entry indicates the number of occurrences of a specific word in pages originating from a specific university. Although both matrices refer (approximately) to the same set of words, the statistics in each matrix is entirely different. Hence, if we are interested in compressing the word-variable, there are two natural choices for the *relevant* variable. Each choice will yield different results, as we show below.

Following standard pre-process steps [1] we had two count matrices. In M_{topic} we had $7,777$ distinct words with respect to the six topics. In M_{univ} we had $9,840$ distinct words with respect to the four universities. Applying direct normalizations we ended up with two (estimated) joint probabilities. In the first one, $p(w, c_{topic})$ we had $|\mathcal{W}| = 7,777$, $|\mathcal{C}_{topic}| = 6$. In the second one, $p(w, c_{univ})$ we had $|\mathcal{W}| = 9,840$, $|\mathcal{C}_{univ}| = 4$.

For both matrices we decided to compress W into a new variable, denoted by T_{topic} and T_{univ}, respectively. For simplicity we decided to consider "flat" solutions (rather than a hierarchy of solutions), and in particular we set $|\mathcal{T}_{topic}| = |\mathcal{T}_{univ}| = 10$. Since this setting already implies a significant compression, we were able to take $\beta^{-1} = 0$, thus to remain focused on maximizing the relevant information terms, $I(T_{topic}; C_{topic})$ and $I(T_{univ}; C_{univ})$. Consequently, the natural choice was to use the sIB algorithm (since setting $\beta^{-1} = 0$ for the iIB algorithm causes numerical difficulties). For each matrix we performed ten restarts using ten different random initializations, and eventually chose the solution which maximized the relevant information.

For $p(w, c_{topic})$, with only ten clusters we got $I(T_{topic}; C_{topic}) \approx 0.14$, which is about 68% of the original information. For $p(w, c_{univ})$, with the same number of clusters, we got $I(T_{univ}; C_{univ}) \approx 0.09$, which is about 74% of the original information. That is, for both choices of the relevant variable, a significant compression implies only a rather minor loss of relevant information.

We further considered the clusters extracted in each case. We sorted all ten clusters in T_{topic} by their contribution to $I(T_{topic}; C_{topic})$, given by

$$I(t_{topic}) \equiv p(t_{topic}) \sum_{c_{topic}} p(c_{topic} \mid t_{topic}) \log \frac{p(c_{topic} \mid t_{topic})}{p(c_{topic})} . \tag{4.1}$$

For each of the five most informative clusters, we present in the upper part of Table 4.1 the five most probable words, which are the words that maximize $p(w \mid t_{topic})$. Clearly, each of these clusters is predictive of one of the values of C_{topic}, or in other words, the partitioning of $\mathcal{W}$ is informative about the different topics in the data.

We perform an identical analysis for the ten clusters in T_{univ}. The results are presented in the lower part of Table 4.1. As expected, the partition of $\mathcal{W}$ is entirely different. In particular, this partition is predictive of the values of C_{univ}, namely about the different sources of the WWW-pages in those data.

4.2 aIB with finite β for non-balanced clustering

In many applications, clustering is used as a tool for analyzing the properties of a large collection of objects. For example, in gene expression data analysis (see, e.g., [28]), one might be interested in clustering on the order of 10^4 genes, according to their expression patterns. Although in some cases clustering the genes into a large number of clusters might be useful, clearly the analysis of, e.g., $\approx 10^3$ different clusters is very time consuming, and in fact not feasible in some cases. Hence, clustering the data into a relatively small number of, e.g., ten clusters, is desirable. However, in this case, a "balanced" clustering solution (which is a typical result for a standard clustering method) will yield around $\approx 1,000$ genes in each cluster. Therefore, considering each and every object in a specific cluster might be also too demanding.

[1] We used *rainbow* software [52] for this pre-process. Specifically we followed the steps suggested in *http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/*, from which the data are also available on-line. To avoid too high dimensionality, we further ignored in each matrix words with fewer than ten occurrences.

Table 4.1: Results for word clustering with respect to different target variables, over the "4 Universities Data Set". The first five rows describe the results for compressing the words while preserving the information about the different topics in the corpus. Results are presented for the five clusters with the highest contribution to $I(T_{topic}; C_{topic})$ (see Eq. (4.1)). The first column indicates the value $t_{topic} \in T_{topic}$. The second column indicates the value of C_{topic} for which $p(c_{topic} \mid t_{topic})$ is maximized, where this maximizing value is given in the next column. The last column presents the ten most probable words in this cluster, ordered by $p(w \mid t_{topic})$. ("<time>" stands for a string in the page representing the time of day, and '\$' stands for any digit character.) The last five rows in the table present the same analysis for compressing the words while preserving the information about the different sources of the pages. Clearly, changing the relevance variable yields an entirely different partition of $\mathcal{W}$.

T value	Most probable relevant value	$p(c \mid t)$	Most probable members
$t^{(1)}_{topic}$	*'Course'*	0.83	*<time>, will, course, class, your, homework, lecture, hours, assignment, assignments.*
$t^{(2)}_{topic}$	*'Student'*	0.69	*my, am, me, working, cornell, personal, austin, stuff, resume, ve.*
$t^{(3)}_{topic}$	*'Faculty'*	0.72	*professor, conference, he, acm, international, ieee, pp, his, journal, member.*
$t^{(4)}_{topic}$	*'Course'*	0.53	$$, $, $$..., *be, programming, office, not, postcript, fall, cs*$$$.
$t^{(5)}_{topic}$	*'Project'*	0.33	*system, parallel, group, based, our, performance, applications, laboratory, high, real.*
$t^{(1)}_{univ}$	*Washington*	0.94	*washington, cse, cse*$$$, *seattle, emacs, wa, sieg, ladner, pst, tompa.*
$t^{(2)}_{univ}$	*Texas*	0.89	*austin, texas, utexas, ut, qualitative, hello, ans, tx, mooney, inductive, acquisition.*
$t^{(3)}_{univ}$	*Wisconsin*	0.95	*wisc, madison, wisconsin, ece, wi, const, char, scores, shore, stl.*
$t^{(4)}_{univ}$	*Cornell*	0.85	*cornell, video, slide, rivl, ithaca, ny, upson, audio, hours, mpeg.*
$t^{(5)}_{univ}$	*Texas*	0.33	*university, systems, system, research, software, language, based, pages, group, sciences.*

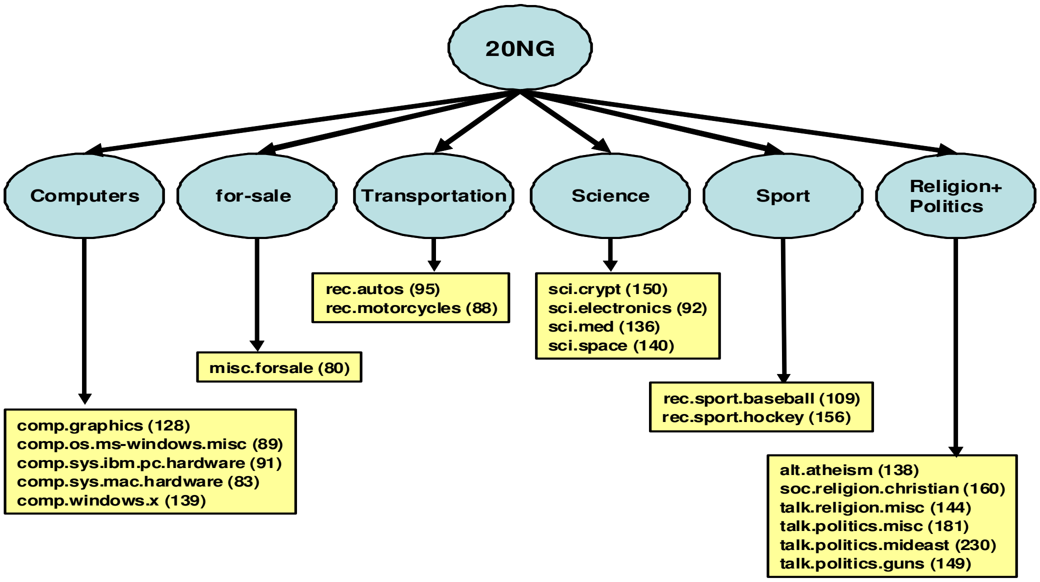

Figure 4.1: Details of the 20NG corpus. The collection contains about $20,000$ messages evenly distributed among 20 different UseNet discussion groups, which define 20 different textual categories (or "topics"). These categories are presented in the figure in a hierarchical manner, which was done manually for purposes of presentation. For each category we indicate the average number of words in its messages, *after* the pre-process described in the text. As noted in [66], a small amount of these messages (less than 5%) are present in more than one group, hence it might be considered as a multi-labeled corpus. However, we ignored this observation in the experiments reported in this thesis.

Bearing this in mind, in some circumstances one might be interested in *non-balanced* clustering solutions. In this case, the number of clusters is rather small, but still most of the clusters are relatively compact, consisting of a small number of objects. It turns out that applying this idea in our context is straightforward. Recall that for "hard" clustering, the compression-information, $I(T;X)$ is simply the entropy of the compression variable, $H(T)$ (see Section 3.3.1). Therefore, using the sIB or the aIB algorithm with a *finite* β value, yields a clustering solution which aims at maximizing the relevant information while minimizing this entropy. Since the entropy decreases as the distribution drifts away from the uniform (most balanced) distribution, finite (small) β values will yield non-balanced clustering solutions. A typical result will consist of one big cluster and additional much smaller clusters. Since we are also concerned with maximizing the relevant information, the values that are merged into the big cluster are usually the less informative ones. In this sense, reducing β is equivalent to inducing a "noise filter", that leaves only the most informative features of X in specific (compact) clusters.

To demonstrate this effect we used a subset of the 20-newsgroups (20NG) corpus, collected by Lang [47]. This collection contains about $20,000$ messages evenly distributed among 20 UseNet discussion groups, some of which have very similar topics (see Figure 4.1 for the details). In this application we concentrated on the $4,000$ messages taken from the four "science" discussion groups: *'sci.crypt', 'sci.electronics', 'sci.med'* and *'sci.space'*. We removed all file headers, leaving the body and subject line only for each message. After lowering upper case characters, uniting all digits into one symbol, ignoring non alpha-numeric characters and removing stop-words and words that occurred only once, we had a counts matrix of $25,896$ distinct words versus $4,000$ documents. Normalizing by the total counts we obtained (an estimate of) a joint distribution, $p(w,d)$, which is the probability that a random word position is equal to $w \in \mathcal{W}$ and at the same time the document is $d \in \mathcal{D}$. To avoid overly high dimensionality we further sorted all words by their contribution to $I(W;D)$, given by $I(w) \equiv p(w) \sum_d p(d \mid w) \log \frac{p(d|w)}{p(d)}$ and selected the top $2,000$. [2] After re-normalization we ended up with a joint distribution, $p(w,d)$ with $|\mathcal{W}| = 2,000,\ |\mathcal{D}| = 4,000$.

Since we were interested to see the effect of reducing β for different resolutions, we applied the aIB algorithm to this joint distribution, using $\beta^{-1} = 0$ and $\beta^{-1} = 0.1$. In the lower right panel of Figure 4.2 we see the two (approximated) normalized relevance-compression curves that were obtained by the algorithm. The horizontal axis corresponds to the normalized compression-information, $\frac{H(T_w)}{H(W)}$. The vertical axis corresponds to the normalized relevant information, $\frac{I(T_w;D)}{I(W;D)}$. Clearly, constraining the compression term as well (by taking a finite β) provides a better (higher) curve. Specifically, for any given level of compression, for $\beta^{-1} = 0.1$, aIB preserves more relevant information about D than for $\beta^{-1} = 0$.

Further considering the results for $|\mathcal{T}_w| = 20$, we see that as expected, the prior probability $p(t_w)$ is much more balanced for $\beta^{-1} = 0$. Specifically, in this case $H(T_w) \approx 2.84$ while for $\beta^{-1} = 0.1$ we get $H(T_w) \approx 2.11$. At the same resolution, the reduction in relevant information is minor: From $I(T_w;D) \approx 0.56$ to ≈ 0.54, respectively. Alternatively we may consider the cluster sizes, i.e, the number of words that were assigned to each cluster. Again, as expected, for $\beta^{-1} = 0.1$ we see that the solution consists of one big cluster (with almost half of the words), and additional much smaller ones. In Figure 4.2 we present the centroids, $p(d \mid t_w)$ for this big cluster and for some of the smaller clusters. Clearly the small clusters are more informative about the structure of D. Moreover, considering the words in each cluster we see that indeed the words in the big cluster are less informative about this structure. Alternatively, words that "passed" the "β-filter" form more informative clusters.

As mentioned at the beginning of this section, a natural application for non-balanced clustering is in gene expression data analysis. However, this application is reserved for future research.

4.3 dIB for "soft" word clustering

In the two previous sections we applied the aIB and the sIB algorithms for "hard" clustering of words. However, as we already mentioned, in our context (for finite β values) "hard" partitions are typically sub-optimal (see Section 3.5). Moreover, a signal of natural language is stochastic in nature. In particular, words may have different meanings, which creates the need to be able to assign a word to different clusters corresponding to the different word senses (see, for example, [60]).

In this section we address this issue by first applying the dIB algorithm to the (full) 20NG corpus. After the same pre-process described in the previous section we got a count matrix of $|D| = 19,997$ documents versus $|W| = 74,000$ distinct words. By summing the counts of all the documents in each class (based on the document labels) and applying simple normalization, we extracted out of this matrix an estimate of a joint distribution, $p(w,c)$, of words versus textual categories (topics) with $|\mathcal{C}| = 20$. We further sorted all words by their contribution to $I(W;C)$ (given by $p(w) \sum_c p(c \mid w) \log \frac{p(c|w)}{p(c)}$) and selected the 200 most informative ones (which capture about 15% of the original information). After re-normalization we ended up with a joint distribution with $|\mathcal{W}| = 200,\ |\mathcal{C}| = 20$.

Given this joint distribution we applied the dIB algorithm to form a hierarchy of word clusters, T_w. The implementation details were as follows. The rate of increasing β was defined through $f_\beta = (1+\varepsilon_\beta)\beta,\ \varepsilon_\beta = 0.001$. The parameter used for detecting splits was defined as $d_{min} = \frac{1}{\beta}$, i.e., as β increases the algorithm becomes more "liberal" in declaring cluster splits. The scaling factor for the stochastic duplication was set to $\alpha = 0.005$.

In Figure 4.3 we present the extracted bifurcating "tree" that traces the solutions at different β values. In each level, every cluster $t_w \in \mathcal{T}_w$ is represented by the four words, $w \in \mathcal{W}$, that maximize the membership probability, $p(t_w \mid w)$. The numbers below each bifurcation indicate the corresponding β value for which the

[2] Note that this feature selection scheme does *not* use the document class-labels.

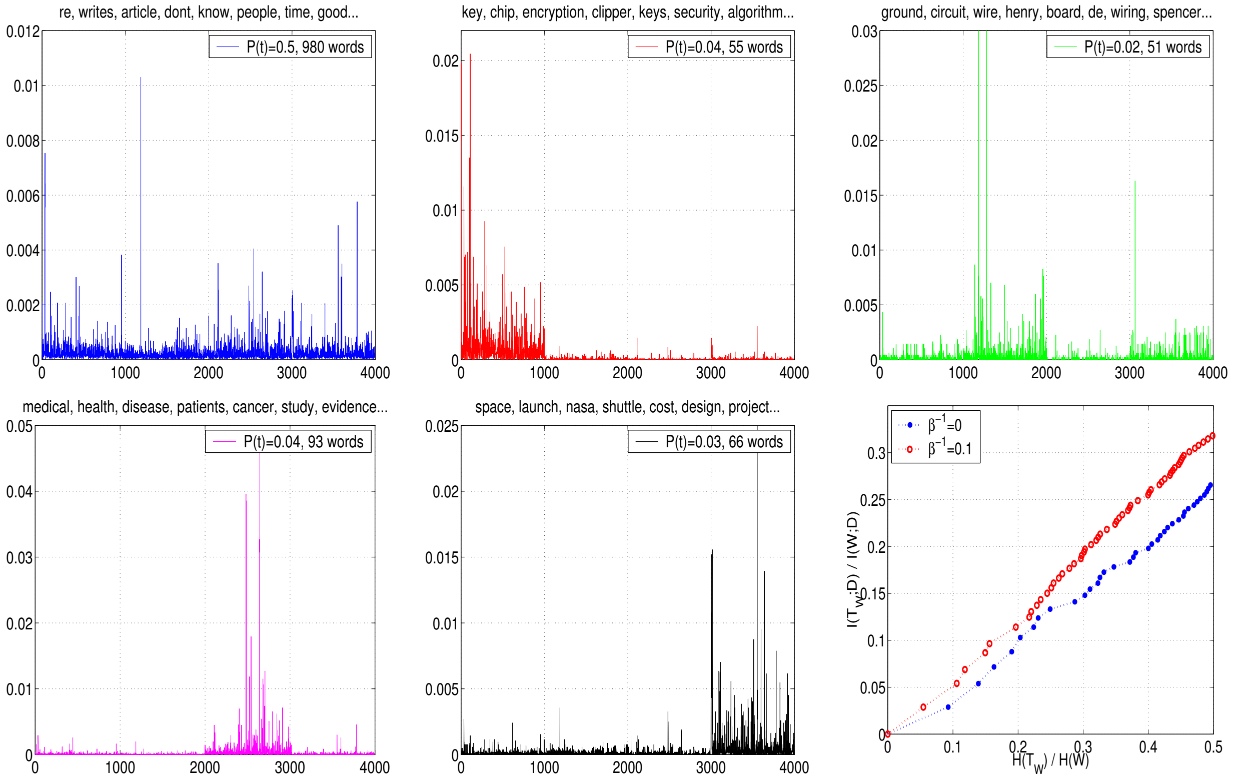

Figure 4.2: aIB results for non-balanced clustering with $\beta^{-1} = 0.1$, $|T_w| = 20$. The first five figures present $p(d \mid t_w)$ for five word clusters, $t_w \in \mathcal{T}_w$. Documents $1 - 1000$ belong to the *sci.crypt* category, $1001 - 2000$ to *sci.electronics*, $2001 - 3000$ to *sci.med* and $3001 - 4000$ to *sci.space*. In the title of each panel we present the words that maximized $p(w \mid t_w)$ in each cluster. The "big" cluster (upper left panel) is clearly less informative about the structure of D than the smaller clusters. In the lower right panel we see the two normalized relevance-compression curves. Given some compression level, for $\beta^{-1} = 0.1$ aIB preserves more relevant information about D than for $\beta^{-1} = 0$.

Table 4.2: "Disambiguated" words in the 20NG word-category data, based on the dIB results for $|\mathcal{T}_w| = 20$. For each word, $w \in \mathcal{W}$ we present all clusters with $p(t_w \mid w) > 0.05$. The first column indicates the word and the second column presents its membership probabilities. For each cluster, $t_w \in \mathcal{T}_w$ we indicate in the third column the category for which $p(c \mid t_w)$ is maximized, where this maximizing value is given in parentheses. In the last column we present the five words that maximize $p(t_w \mid w)$ (where '\$' stands for any digit character).

W **value**	$P(t_w \mid w)$	**Typical prediction**	**Typical members**
speed	0.49	*comp.graphics* (0.16)	*code, version, file, screen, ftp*
	0.39	*rec.sport.hockey* (0.26)	*leafs, nhl, hockey, season, team*
	0.07	*comp.sys.ibm.pc.hardware* (0.44)	*scsi\$, ide, scsi, bios, controller*
killed	0.62	*talk.politics.misc* (0.10)	*believe, say, science, people, life*
	0.26	*talk.politics.guns* (0.39)	*firearms, atf, gun, guns, batf*
	0.11	*talk.politics.mideast* (0.84)	*armenians, arabs, armenian, israeli, armenia*
price	0.70	*rec.sport.hockey* (0.26)	*leafs, nhl, hockey, season. teams*
	0.23	*comp.graphics* (0.16)	*code, version, file, screen, ftp*
rights	0.57	*talk.politics.guns* (0.39)	*firearms, atf, gun, guns, batf*
	0.43	*talk.politics.misc* (0.10)	*believe, say, science, people, life*
religious	0.73	*talk.politics.misc* (0.10)	*believe, say, science, people, life*
	0.25	*soc.religion.christian* (0.40)	*sin, christianity, christians, christ, bible*
truth	0.78	*soc.religion.christian* (0.40)	*sin, christianity, christians, christ, bible*
	0.22	*talk.politics.misc* (0.10)	*believe, say, science, people, life*
manager	0.91	*comp.graphics* (0.16)	*code, version, file, screen, ftp*
	0.06	*rec.sport.hockey* (0.26)	*leafs, nhl, hockey, season, team*
earth	0.92	*talk.politics.misc* (0.10)	*believe, say, science, people, life*
	0.06	*sci.space* (0.79)	*spacecraft, shuttle, orbit, launch, moon*

split occur. As can be seen in the figure, as β increases additional splits emerge, revealing a finer structure of the data, which is indeed informative about the topics of the corpus. Specifically, as the resolution (i.e., the number of clusters $|\mathcal{T}_w|$) increases, additional clusters become more specific in their prediction, which is reflected by the semantic relation of their members to one of the topics in the corpus. Additionally, note that the splits typically divide into several "groups", where each "group" occurs in a small range of β values (see also Figure 4.4).

Considering a specific level in this hierarchy of solutions, for $|\mathcal{T}_w| = 20$, we see that for words that are relevant to predicting more than a single topic, the mapping $P(t_w \mid w)$ is indeed stochastic. For example, the word *'speed'* is assigned with a probability of 0.49 to a cluster which is predictive of the "computer topics" and with a probability of 0.39 to a different cluster which is predictive of the "sport topics". Hence, since *'speed'* is disambiguated with respect to our relevant variable, the algorithm assigns it to more than one cluster. Other examples are given in Table 4.2.

Although the above examples are intuitively clear, our objective performance measure is how well the dIB algorithm optimizes the trade-off defined by the IB-functional. To examine this, in the left panel of Figure 4.5 we present the normalized relevance-compression curve that corresponds to the dIB solutions. For comparison we also present the relevance-compression curve of the "hard" solutions extracted by the aIB algorithm for $|\mathcal{T}_w| = 200, 199, \ldots, 1$ (with $\beta^{-1} = 0$). As expected, the stochastic nature of the dIB solutions is reflected in a better curve, where for a given compression level, the dIB solution preserves a higher fraction of the original relevant information.

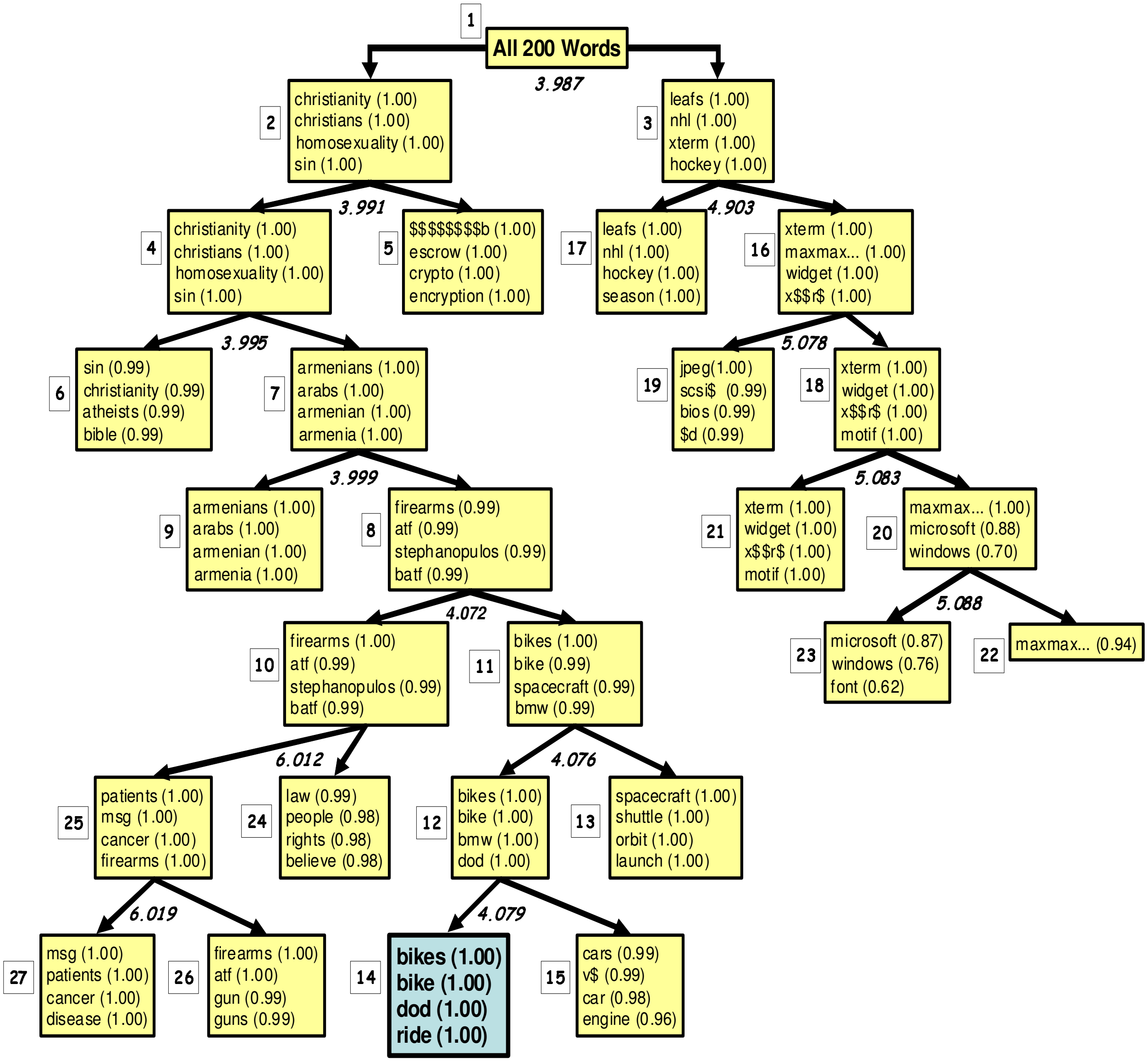

Figure 4.3: dIB results for the 20NG word-category data. Each word cluster is represented by the four words that maximize the membership probability, $p(t_w \mid w)$, where the value of this probability is indicated in parentheses. ('$' stands for any digit character.) The serial numbers on the left of each cluster indicate the order of splits, and the numbers below each split indicate the corresponding β values. As β increases, more clusters are needed to attain the required minimal level of relevant information. As a result, clusters bifurcate into more specific ones that are predictive of specific topics in the corpus. This is reflected by the semantic relationship of the members in each cluster to one of the topics in the corpus. For example, considering the emphasized cluster in the lower level, we see that its members are semantically related to the *'rec.motorcycles'* category. If we consider the predictions of this cluster over the categories, given by $p(c \mid t_w)$, we see that this probability is indeed maximized (and equals ≈ 0.87) for the related *'rec.motorcycles'* category. Note that after each increment of β, all the membership probabilities, $p(t_w \mid w)$ are updated (where the previous solution is only used for the initialization). Hence, the values of $p(t_w \mid w)$ in the figure reflect the values right after the split, and might be different after further splits. Moreover, the extracted hierarchy does not necessarily construct a tree, and in principle values that are assigned to some branch might be assigned later on to other branches.

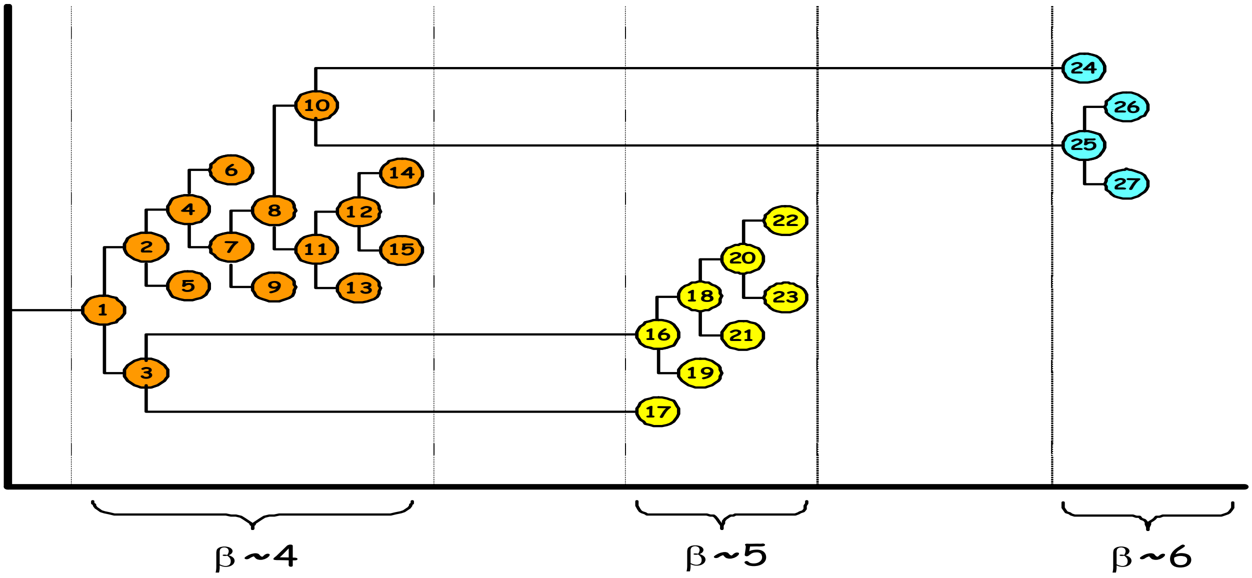

Figure 4.4: dIB results for the 20NG word-category data. Each circle corresponds to a cluster in Figure 4.3 with the corresponding serial number. Interestingly, the splits typically divide into several "groups", where each "group" occurs on a small range of β values (the exact β values are indicated in Figure 4.3).

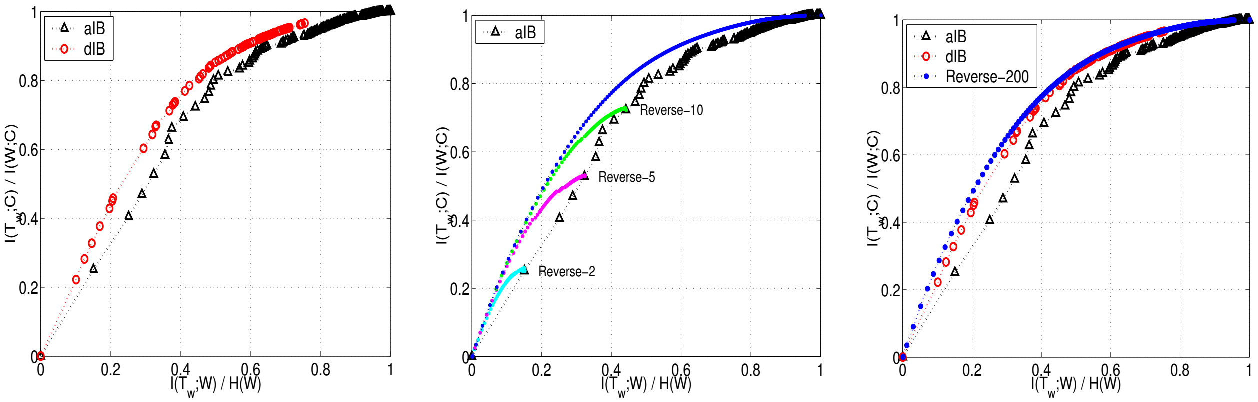

Figure 4.5: Estimates of normalized relevance-compression curves for the 20NG word-category data. The original relevant information is $I(W;C) \approx 0.84$, and the original compression-information is $H(W) \approx 4.34$. **Left:** Comparison of the curves extracted by dIB and aIB. The stochastic nature of the dIB solutions results in a higher (i.e., "better") curve. **Middle:** Reverse-annealing curves, where the "hard" aIB solutions are used as the initialization. Note that all these curves converge to the same "envelope" curve, which is the curve we find by initializing the reverse-annealing process at the trivial "hard" solution of 200 clusters, where $T_w \equiv W$. This suggests that this "envelope" curve is the globally optimal relevance-compression curve for these data. **Right:** Comparison of the "envelope" reverse-annealing curve with the dIB curve. Reverse-annealing yields a slightly higher curve, suggesting that this process, which is not required to detect clusters bifurcations, is more robust to the presence of local optima.

4.3.1 Reverse annealing for estimating the relevance-compression function

An alternative way of extracting a sequence of "soft" solutions is through the process of reverse-annealing discussed in Section 3.6. Recall that in this scheme we start from a "hard" solution that can be (approximately) represented through Eq. (2.16) with a large enough β. We now gradually *decrease* β and for each new β value we use the iIB algorithm to extract a locally optimal solution, where the previous solution is used as the initialization.

We applied this scheme to the same data where we used the "hard" aIB solutions as platforms for recovering sequences of (perhaps locally) optimal "soft" solutions. [3] In the middle panel of Figure 4.5 we present several normalized relevance-compression curves, corresponding to these sequences of solutions. Interestingly, all these curves (including curves that are not presented to ease the presentation) converged to the same "envelope" curve, which is the curve we got by initializing the reverse-annealing process at the trivial "hard" solution of 200 clusters, where $T_w = W$.

These results match the theoretical analysis in Section 2.3 (in particular note the similarity to the illustration in Figure 2.6). The curves that resulted from a "hard" solution with $|\mathcal{T}_w| < |\mathcal{W}|$ correspond to the sub-optimal relevance-compression curves that are further constrained by the cardinality, $|\mathcal{T}_w|$. Apparently, each of these curves converges to the "envelope" curve at some point which defines a critical β value, where higher β values require more clusters to remain on the (globally) optimal curve.

In the right panel of Figure 4.5 we further compare the "envelope" reverse-annealing curve with the curve we got through dIB. As seen in the figure, the differences are minor, where there is a small (although consistent) advantage in favor of the reverse-annealing curve. Note that while in the dIB case the number of representative clusters changes along the curve, in the reverse-annealing case this number is fixed (and equals 200 in this case). However, as we continue to decrease β, the *effective* number of clusters decreases. That is, different representatives are collapsing to the same $p(t_w \mid w)$ distribution, and by that reduce (compress) $I(T_w; W)$, as required.

Although we cannot guarantee that the reverse-annealing "envelope" curve is indeed the globally optimal relevance-compression curve, the fact that empirically all the reverse-annealing sub-optimal curves (with $|\mathcal{T}_w| < |\mathcal{W}|$) are converging to it supports this conjecture. If this is the case, we can conclude that the collection of these reverse-annealing curves provides a full characterization of our input, in terms of compression versus preservation of relevant information.

4.4 iIB and sIB sensitivity to local optima

As already discussed in Section 3.1.1, even for a given β, the IB-functional in general have multiple local optima. Therefore, given a joint distribution $p(x, y)$, different initializations of the iIB or the sIB algorithms will typically converge to different locally optimal solutions. An important question is to characterize how sensitive these algorithms are to the presence of local optima. In this section we show that this sensitivity is fairly low if the joint distribution is well estimated and the problem has a "natural" solution.

To address this issue under clear (and controlled) conditions, we use synthetic data in this application. To generate the data we used a standard multinomial mixture model. The model and its relationships to the IB method are discussed in detail in Appendix A. For completeness we repeat here the description of the generative process that underlies it. We assume that y takes on discrete values and sample it from a multinomial distribution $\theta(y|c(x))$, where $c(x)$ denotes the (hidden) class label of some $x \in \mathcal{X}$. We further assume [44] [61] that there can be multiple observations of y corresponding to a single x but they are all sampled from the same multinomial distribution. Therefore, the generative process can be described as follows.

- For each $x \in \mathcal{X}$ choose a unique class label $c(x)$ by sampling from $\pi(c)$.

[3] The β values we tested with were $100,\ 99.5,\ \ldots,\ 50.5,\ 50,\ 49.9,\ \ldots,\ 0.2,\ 0.1,\ 0.099,\ \ldots,\ 0.002,\ 0.001$.

- For $k = 1 : N$
  - Choose $x \in \mathcal{X}$ by sampling from $\gamma(x)$.
  - Choose $y \in \mathcal{Y}$ by sampling from $\theta(y|c(x))$ and increase $n(x, y)$ by one.

The estimated joint distribution is then given by $\hat{p}(x, y) = \frac{n(x,y)}{N}$. The parameters of the model, $\pi(c),\ \gamma(x)$ and $\theta(y \mid x)$ correspond to the class prior probability, X prior probability and the class conditional (hidden) probabilities, respectively.

As the sample size N increases, the estimates of $p(y \mid c)$, given by every $\hat{p}(y \mid x),\ s.t.\ c(x) = c$ improve. Therefore, roughly speaking, if the classes are sufficiently different from each other, for $N \to \infty$, the problem of clustering X values has a "natural" unique solution, which is the partitioning that corresponds to the "correct" partition, given by the hidden class labels. In this sense we may say that the convexity of the problem increases as N increases. Therefore, in particular we expect that for a large enough N the sensitivity of the iIB and sIB algorithms to the initialization would be minor.

We used a *real* document-word count matrix to estimate the model parameters. Specifically we used the *Multi*10_1 subset of the 20NG corpus (which is described in detail in Section 4.5). This subset consists of 500 documents randomly chosen from ten different discussion groups. The corresponding count matrix refers to the $2,000$ words that maximize the information about the documents. Therefore, we have $|\mathcal{D}| = 500,\ |\mathcal{W}| = 2,000,\ |\mathcal{C}| = 10$. Using the document class labels it is straightforward to get an estimate for the model parameters. Based on these estimates and on the generative model we produced several different count matrices for different sample sizes, ending up with different estimates of the joint distribution, $p(d, w)$. For each of these estimates we first applied the iIB algorithm to cluster the documents into ten clusters, that is we took $|\mathcal{T}_d| = 10$. We further set $\beta = 20$ and applied 100 different initializations for each input matrix, yielding 100 (locally) optimal solutions for each value of N.

In Figure 4.6 we present the results for $N = 50,000,\ 200,000,\ 500,000$. In the upper panel we present all solutions in the relevance-compression plane. As the sample size N increases, the scatter of the solutions in this plane becomes more concentrate, as predicted. Considering the same solutions in the normalized relevance-compression plane we see the same phenomenon. Note that in this plane, as N increases, the solutions found by the algorithm typically preserve a higher *fraction* of the original ("empirical") relevant information. This is a direct result of a well known effect of an upper bias in the estimate of the empirical mutual information due to a small sample size (see, e.g., [84]). Namely, our estimates to the original relevant information, $I(D; W)$ are upper biased for lower N. Therefore, our estimations of $\frac{I(T_d;W)}{I(D;W)}$ are typically biased downward (i.e., provide a "worst-case" estimation) for small sample sizes. We further discuss this issue in Section 6.1.

In the lower panel of Figure 4.6 we consider the *Precision* of each solution given by its correlation to the "correct" partition (this term is defined explicitly in Section 4.5.2). Considering the histogram of the 100 precision values for each N, we see that as N increases the precision is (significantly) improving. That is, for larger N, more solutions are well correlated with the "correct" partitioning of the (synthetic) documents. Moreover, in this histogram as well we see that the scatter of the solutions is decreased as N increases, which also implies that the number of different locally optimal solutions is decreasing.

We further applied the sIB algorithm to different estimates of $p(d, w)$. We used the same setting as in the iIB case, that is $|\mathcal{T}_d| = 10,\ \beta = 20$ and applied 100 different initializations for each input matrix, yielding 100 (locally) optimal solutions for each value of N. Recall that the solution space of sIB consists of "hard" clustering solutions only, hence it is dramatically smaller than the full solution space which is explored by iIB. Additionally, the definition of local optimality here is with respect to a different optimization routine. Namely, convergence is declared when no more single (and discrete) assignment updates can improve the IB-functional.

In Figure 4.7 we present the results for this algorithm. It was found to be significantly less sensitive to the initialization than the iIB algorithm. In fact, only for a very small sample size this sensitivity was clearly evident, hence in this case we present the results for for $N = 5,5000,\ 50,000,\ 200,000.$ [4] In the upper panel we present all solutions in the relevance-compression plane and in the middle panel we present the same solutions in the normalized relevance-compression plane. Clearly, for all sample sizes the scatter of the solutions is much smaller as opposed to the iIB results. Moreover, the sIB solutions typically attain higher relevant information values, but also higher compression-information values, which is probably due to the fact that sIB extracts "hard" clustering solutions.

Although in the relevance-compression plane even for $N = 5,500$ there is apparently no significant difference between the 100 different solutions, while considering the precision histogram (lower panel) the picture is quite different. Specifically, different initializations yield different solutions with a relatively wide range of precision levels (very similar to what we got for iIB with a ten-times larger N). As the sample size increases, more initializations converge effectively to the same solution. In particular, for $N = 200,000$, we find that 83 out of the 100 initializations converge to the same solution in terms of the IB-functional (upper right dot in the figure). Moreover, for all these solutions the precision is exactly 89.6%, also implying that all these solutions correspond effectively to a single solution. It is reasonable to assume that this solution is in fact the *globally* optimal solution of the IB-functional (for this sample size and β value), among all the possible "hard" solutions.

4.5 sIB and aIB for unsupervised document classification

In the last section of this chapter we investigate a more practically oriented application. Unsupervised document clustering is a central problem in information retrieval. Possible applications include use of clustering for improving retrieval [85], and for navigating and browsing large document collections [27, 43, 89]. Several recent works suggest using clustering techniques for *unsupervised* document classification [30, 73, 77]. In this task, we are given a collection of unlabeled documents and attempt to find clusters that are highly correlated with the true topics of the documents. This practical situation is especially difficult since no labeled examples are provided for the topics, hence unsupervised methods must be employed.

In this section, following [74], we address this task using the sIB and the aIB algorithms and provide a thorough comparison of their performance with other clustering techniques. In our evaluation, on small and medium size real world corpora, the sIB algorithm is found to be consistently superior to all the other clustering methods we examine, typically by a significant margin. Moreover, the sIB results are comparable to those obtained by a *supervised* Naive Bayes classifier. Finally, we propose a simple procedure for trading cluster recall to gain higher precision, and show how this approach can extract clusters which match the existing topics of the corpus almost perfectly. A preliminary theoretical analysis that supports our empirical findings is given in Appendix B

4.5.1 The datasets

Following [30, 77] we used several standard *labeled* datasets to evaluate the different clustering methods. As our first dataset we again used the 20NG corpus [47]. For small-scale experiments we used the nine subsets of this corpus already used in [30, 77]. Each of these subsets consists of 500 documents randomly chosen from several discussion groups (see Table 4.5.1). For each subset we performed the exact same pre-process described in Section 4.2, and further normalized the counts in each document independently (to avoid a bias due to different documents lengths). Thus, we ended up with nine (estimated) joint probabilities, with $|\mathcal{D}| = 500,\ |\mathcal{W}| = 2{,}000,\ p(d) = \frac{1}{|\mathcal{D}|}$.

For a medium scale experiment we used the whole corpus. We again performed the same pre-process and further ignored documents which were left with less than ten word occurrences, ending up with an

[4] For the smallest sample size, to ensure that each "document" will have at least one "word" occurrence, we first sampled a single $y \in \mathcal{Y}$ for every $x \in \mathcal{X}$, and then added $5,000$ samples according to the model, ending up with $N = 5,500$.

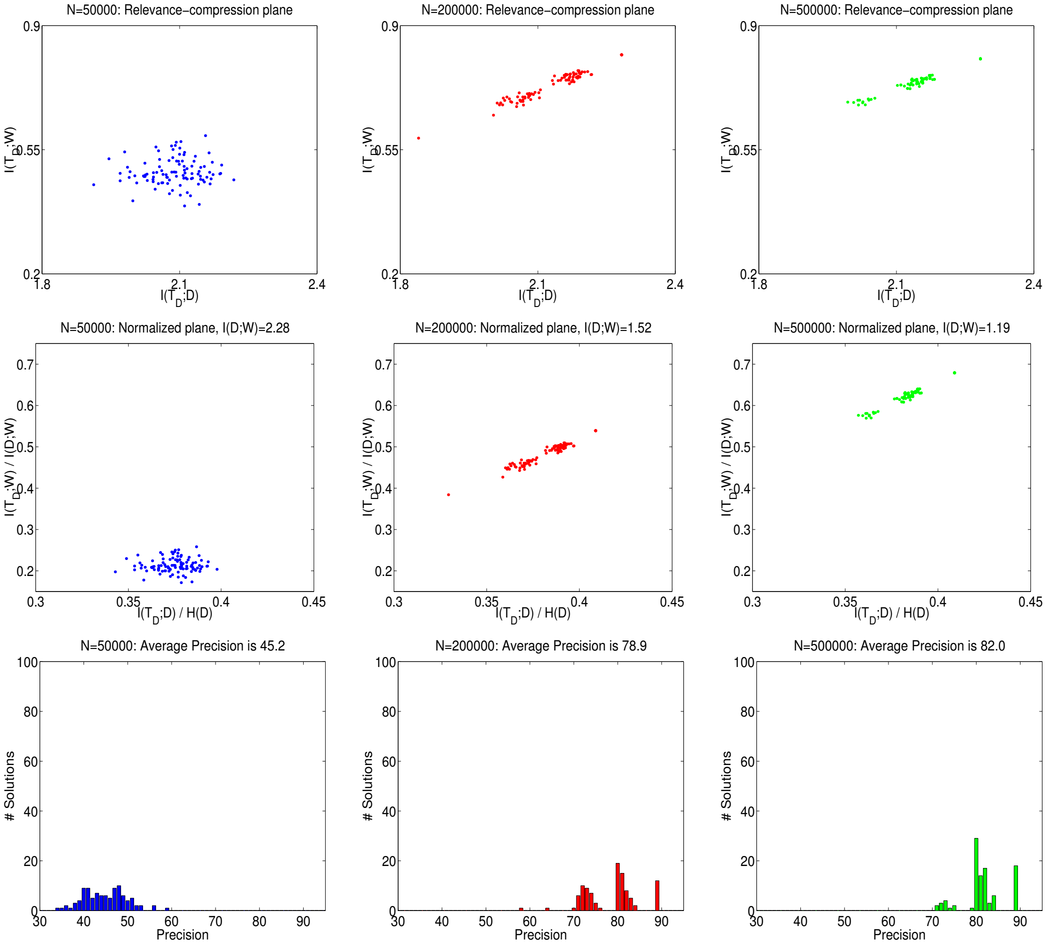

Figure 4.6: iIB results for $N = 50,000,\ 200,000,\ 500,000$. **Upper panel:** Results for 100 different initializations of the iIB algorithm in the relevance-compression plane for different sample sizes. As the sample size increases, the scatter of the solutions becomes more concentrated. **Middle panel:** The same results in the normalized relevance-compression plane. For larger N values the solutions found by the algorithm preserve a higher *fraction* of the original (empirical) relevant information, which is upper biased for small sample sizes. **Lower panel:** Histograms of the 100 precision values for different sample sizes. As the sample size increases the average precision improves. Additionally, more initializations tend to yield the same precision level, which implies less sensitivity to the presence of local optima.

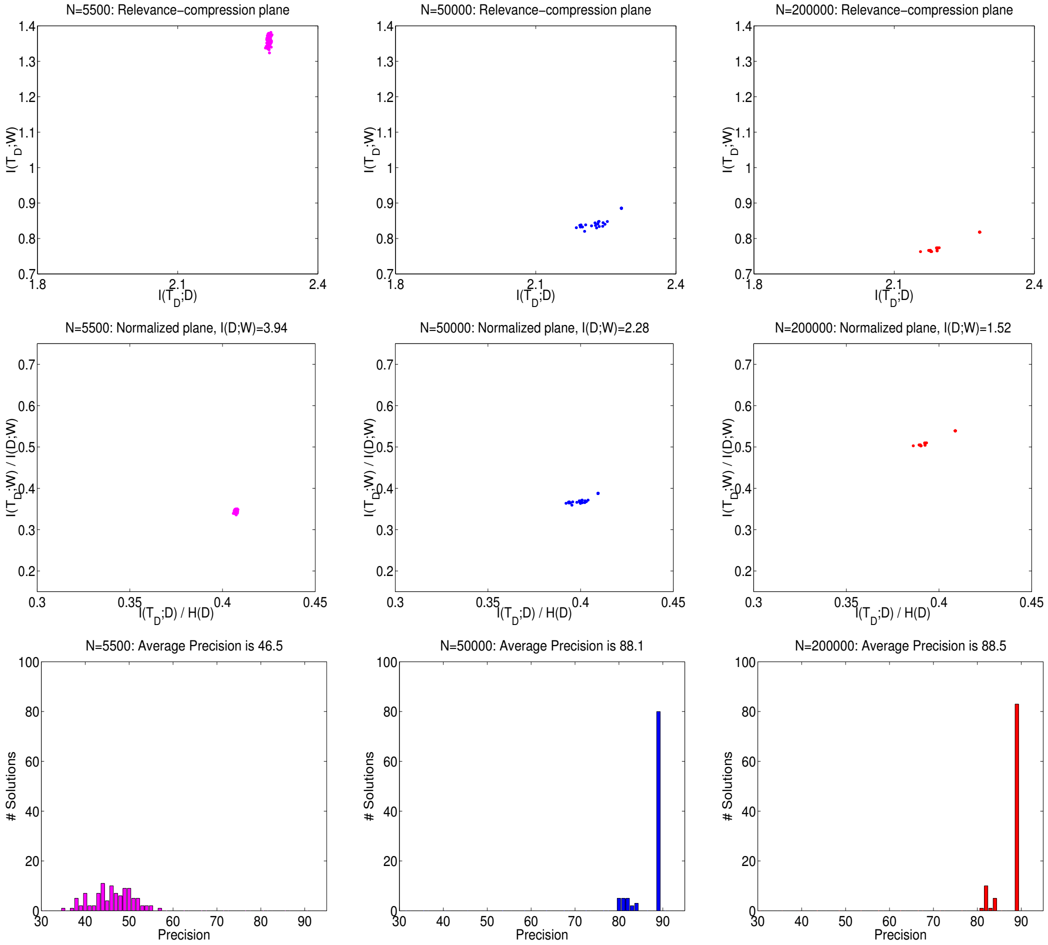

Figure 4.7: sIB results for $N = 5,500,\ 50,000,\ 200,000$. **Upper panel:** Results for 100 different initializations of the sIB algorithm in the relevance-compression plane for different sample sizes. **Middle panel:** The same results in the normalized relevance-compression plane. Clearly, the scatter of the results is much lower than the iIB results. Additionally, the sIB solutions typically attain higher relevant and compression information values. **Lower panel:** Histograms of the 100 precision values for different sample sizes. For $N = 200,000$, there are 83 solutions with the same precision of 89.6% and the same value of the IB-functional (upper right dot in the upper right figure). This implies that all these 83 initializations converged effectively to the same optimal solution, which is presumably the global optimum of the IB-functional in this setting, among all the possible “hard” solutions.

Table 4.3: Datasets details for the nine small subsets of the $20NG$ corpus. For example, for each of the three *Binary* datasets we randomly chose 500 documents, evenly distributed between the discussion groups *talk.politics.mideast* and *talk.politics.misc*. This resulted in three document collections, *Binary*$_1$, *Binary*$_2$ and *Binary*$_3$, each of which consisted of 500 documents.

Dataset	Newsgroups included	#docs per group	Total #docs
Binary$_{1,2,3}$	*talk.politics.mideast,* *talk.politics.misc.*	250	500
Multi$5_{1,2,3}$	*comp.graphics, rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast.*	100	500
Multi$10_{1,2,3}$	*alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.med, sci.electronics, sci.space, talk.politics.guns.*	50	500

estimated joint distribution, $p(w, d)$ with $|\mathcal{D}| = 17,446$, $|\mathcal{W}| = 2,000$, $p(d) = \frac{1}{|\mathcal{D}|}$. We constructed two different tests over these data. First we measured our performance with respect to all the 20 different classes. Additionally we applied an easier test where we measured our performance with respect to ten meta-categories in this corpus. [5] We term these two tests *NG20* and *NG10* respectively.

As an additional medium scale test we used the $10,789$ documents of the ten most frequent categories in the Reuters-21578 corpus [6] under the ModApte split. After the same pre-process we got an estimated joint distribution with $|\mathcal{D}| = 8,796$, $|\mathcal{W}| = 2,000$, $p(d) = \frac{1}{|\mathcal{D}|}$.

As the last medium scale test we used a subset of the new release of the Reuters-2000 corpus. Specifically we used the $22,498$ documents of the ten most frequent categories in the ten first days of this corpus (last ten days in August 1996). After the same pre-process (except for not uniting digits due to a technical reason), we ended up with an estimated joint distribution with $|\mathcal{D}| = 22,463$, $|\mathcal{W}| = 2,000$, $p(d) = \frac{1}{|\mathcal{D}|}$. Note that these two last Reuters corpora are multi labeled.

4.5.2 The evaluation method

As our evaluation measures we used micro-averaged precision and recall. To estimate these measures we first assign all the documents in some cluster $t_d \in \mathcal{T}_d$ with the most dominant label in that cluster. [7] Given these uni-labeled assignments we can estimate for each category $c \in \mathcal{C}$ the following quantities: $A_1(c, T_d)$ defines the number of documents correctly assigned to c (i.e., their true label sets include c), $A_2(c, T_d)$ defines the number of documents incorrectly assigned to c and $A_3(c, T_d)$ defines the number of documents incorrectly not assigned to c. The micro-averaged precision is now defined through (see, e.g, [71]):

$$Prec(T_d) = \frac{\sum_c A_1(c, T_d)}{\sum_c A_1(c, T_d) + A_2(c, T_d)} , \tag{4.2}$$

[5] Specifically we united the five "computer" categories, the three "religion" categories, the three "politics" categories, the two "sport" categories and the two "transportation" categories into five big meta-categories.

[6] Available at *http://www.daviddlewis.com/resources/testcollections/reuters21578/.*

[7] The underlying assumption here is that if the cluster is relatively homogeneous the user will be able to correctly identify its most dominant topic.

while the micro-averaged recall is defined by

$$Rec(T_d) = \frac{\sum_c A_1(c, T_d)}{\sum_c A_1(c, T_d) + A_3(c, T_d)} \ . \tag{4.3}$$

It is easy to verify that if the corpus and the algorithm are both uni-labeled then $Prec(T_d) = Rec(T_d)$, thus for our uni-labeled datasets we will report only $Prec(T_d)$.

As a simplifying assumption we assume that the user is (approximately) aware of the correct number of categories in the corpus. Therefore, for all the unsupervised techniques we measure $Prec(T_d)$ and $Rec(T_d)$ for $|\mathcal{T}_d| = |\mathcal{C}|$. Choosing the appropriate number of clusters is in general a question of model selection which we do not address in this application (see Section 6.2.2 for a discussion).

4.5.3 Other clustering algorithms for comparison

The theoretical analysis in Appendix B suggests that solutions that preserve more relevant information tend to attain higher precision. In our context this means that we should set $\beta^{-1} = 0$ so as to concentrate solely on maximizing $I(T_d; W)$. As discussed in Section 3.5, for this setting the natural choice is to use the sIB and the aIB algorithms. Additionally, under this setting, the merging criterion used by the sIB algorithm is simply

$$\Delta\mathcal{L}(d, t_d) = (p(d) + p(t_d)) \cdot \bar{d}(d, t_d) \ , \tag{4.4}$$

where $d \in \mathcal{D}$ is some document (in a singleton cluster), $t_d \in \mathcal{T}_d$ is some cluster, and $\bar{d}(d, t_d) = JS_\Pi[p(w \mid d), p(w \mid t_d)]$.

An immediate observation is that in principle we can use exactly the same sequential optimization routine of the sIB algorithm (Figure 3.5), while using different choices for $\bar{d}(d, t_d)$. For purposes of comparison we construct several such algorithms.

First, we define $\bar{d}(d, t_d) = D_{KL}[p(w \mid d) \| p(w \mid t_d)]$ and refer to the resulting algorithm as the *'sKL'* algorithm. Second, we use another common divergence measure among probability distributions which is the L_1 norm, given by $\|(p(w) - q(w)\|_1 = \sum_w \mid p(w) - q(w) \mid$. Unlike the JS (and the KL) divergence the L_1 norm satisfies all the metric properties. It is also known to approximate the JS divergence for "similar" distributions [50]. Therefore we define the *'sL1'* algorithm by setting $\bar{d}(d, t_d) = \|p(w \mid d) - p(w \mid t_d)\|_1$. Lastly, we use the standard cosine measure under the vector space model [65]. Specifically we define $\bar{d}(d, t_d) = \langle \hat{d}, \hat{t}_d \rangle$ where $\hat{d}$ is the (*tf*) counts vector of d normalized such that $\|\hat{d}\|_2 = 1$. The centroid $\hat{t}_d$ is defined as the average of all the (normalized) count vectors representing the documents assigned into t_d (again, normalized to 1 under the L_2 norm). Due to this normalization $\langle \hat{d}, \hat{t}_d \rangle$ is simply the cosine of the angle between these two vectors (and is proportional with an opposite sign to $\|\hat{d} - \hat{t}_d\|_2^2$). Note that in this case we update the assignments by merging d into $t_d^{new}(d) = \arg\max_{t_d} \langle \hat{d}, \hat{t}_d \rangle$, i.e., in particular there is no multiplication with the "priors" $p(d),\ p(t_d)$ which are not well defined in this setting. We will term this algorithm *'sK-means'*. We also implemented a standard *parallel* version of this algorithm which we will term *'K-means'*. In this version, given some partition T we re-assign every $d \in \mathcal{D}$ into its closest centroid (under the L_2 norm) and only then re-calculate the centroids. We repeat this process until convergence.

Finally, we also compare our results to the recent Iterative Double Clustering *(IDC)* procedure suggested by El-Yaniv and Souroujon [30]. This method, which is a natural extension of our previous work in [77], uses an iterative double-clustering procedure over documents and words. It was shown in [30] to work extremely well on relatively small datasets, and even to be competitive with a *supervised* SVM classifier trained on a small training set.

One last issue we need to address is how to evaluate different restarts (initializations) of the algorithms. For the sIB we naturally choose the run that maximized $I(T_d; W)$ and report results for it. For other algorithms, we can use their respective scoring function. However, to ensure that this does not lead to poor performance, we choose to present for each of these algorithms the *best* result, in terms of the correlation (precision) to

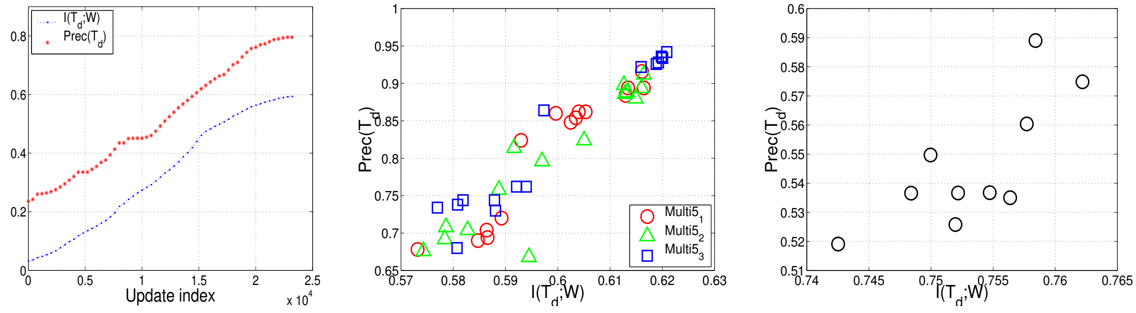

Figure 4.8: **Left:** Progress of $I(T_d;W)$ and $Prec(T_d)$ during the assignment updates of sIB over the *NG10* dataset. **Middle:** Correlation of *final values* of $I(T_d;W)$ and $Prec(T_d)$ for all 15 random restarts of sIB over each of the three *Multi*5 tests. **Right:** Correlation of *final values* of $I(T_d;W)$ and $Prec(T_d)$ for all 10 random restarts of sIB over the *NG*20 test.

the true classification, out of all different restarts. This choice provides an overestimate of the precision of these algorithms, and thus penalizes the sIB algorithm in the comparisons below.

4.5.4 Maximizing information and cluster precision

A first natural question to ask is what differences there are in the performance of the sIB algorithm versus the aIB algorithm in terms of maximizing the relevant information, $I(T_d;W)$. Comparing the results over the nine small datasets (for which running aIB is feasible) we found that sIB (with 15 reatsrts) always extracts solutions that preserve significantly more relevant information than aIB (the improvement is of 17% on the average). Moreover, even if we do not choose the best restart for sIB but compare *all* the 15 random initialization (for every data set) with the aIB results, we found that more than 90% of these runs preserve more relevant information than aIB. These results clearly demonstrate the fact that in contrast to the (greedy) aIB algorithm, the sIB algorithm is always guaranteed to converge to a locally optimal solution of the IB-functional.

The next question we address is whether clustering solutions that preserve more relevant information are better correlated with the real statistical sources (i.e., the categories). In Figure 4.8(a) we present the progress of the relevant information and the precision for a specific restart of sIB over the *NG*10 dataset. We clearly see that while the information increases for every assignment update (as guaranteed by the algorithm), $Prec(T_d)$ increases in parallel. In fact, less than 5% of the updates reduced $Prec(T_d)$. Similar results obtained for all the other datasets.

Lastly, we would like to check whether choosing the restart which maximized $I(T_d;W)$ is a reasonable (unsupervised) criterion for identifying solutions with high precision. In Figure 4.8(b,c) we see the final values of $I(T_d;W)$ versus $Prec(T_d)$ for all the random restarts of sIB over the three *Multi*5 tests and the *NG*20 test. Clearly these final values are well correlated. In fact, in 9 out of our 13 tests the iteration which maximized $I(T_d;W)$ also maximized $Prec(T_d)$, and when it did not the gap was relatively small.

4.5.5 Results for small-scale experiments

In Table 4.4 we present the precision results for the nine small-scale subsets of the 20NG corpus. The results for the *IDC* algorithm are taken from [30]. For all the unsupervised algorithms we applied 15 different random initializations and limited the number of iterations over $\mathcal{D}$ to 30. However, all algorithms except for *sL1* attained full convergence in all 15 restarts and over all datasets after less than 30 loops.

Table 4.4: Micro-averaged precision results over the nine small datasets. In all unsupervised algorithms the number of clusters was taken to be identical with the number of real categories (indicated in parentheses). For *K-means, sK-means, sL1*, and *sKL* the reported precision results are the best results out of all 15 restarts. For sIB the results are for the restart which maximized $I(T_d; W)$. The test set for the *NB* classifier consisted of the same 500 documents in each dataset while the training set consisted of additional 500 documents randomly chosen from the appropriate categories. We repeated this process 10 times and averaged the results.

$Prec(T_d)$	sIB	*IDC*	*sK-means*	*K-means*	aIB	*sL1*	*sKL*	*NB*
Binary$_1$ *(2)*	91.4	85	62.4	65.6	84.0	74.4	50.4	87.8
Binary$_2$ *(2)*	89.2	83	54.6	61.8	59.8	58.0	50.2	85.4
Binary$_3$ *(2)*	93.0	80	63.2	64.0	85.0	76.6	51.8	88.1
Multi5$_1$ *(5)*	89.4	86	47.0	47.4	56.6	51.6	20.6	92.8
Multi5$_2$ *(5)*	91.2	88	47.0	46.0	63.8	45.2	20.6	92.6
Multi5$_3$ *(5)*	94.2	86	57.0	50.4	76.8	52.4	20.6	93.2
Multi10$_1$ *(10)*	70.2	56	31.0	30.8	42.4	34.2	10.4	73.5
Multi10$_2$ *(10)*	63.8	49	32.8	31.0	34.0	31.2	10.0	74.6
Multi10$_3$ *(10)*	67.0	55	33.8	31.4	38.8	31.4	10.2	74.6
Average	**83.3**	**74.0**	**47.6**	**47.6**	**60.1**	**50.6**	**27.2**	**84.7**

To gain some perspective about how hard the classification task is we also present the results of a *supervised* Naive Bayes (*NB*) classifier (see [78] for the details of the implementation). The test set for this classifier consisted of the same 500 documents in each dataset while the training set consisted of additional 500 documents randomly chosen from the appropriate categories. We repeated this process 10 times and averaged the results.

Several results should be noted specifically:

- sIB outperformed all the other unsupervised techniques in all datasets, typically by a significant gap. Given that for the other techniques we present an “unfair” choice of the *best* result (out of all 15 restarts) we see these results as especially encouraging.

- In particular, sIB was clearly superior to *IDC* and aIB which are also both motivated by the IB method. Nonetheless, in contrast to sIB, the specific implementation of *IDC* in [30] is not guaranteed to maximize $I(T_d; W)$ which might explain its inferior performance. We believe that the same explanation holds for the inferiority of aIB.

- sIB was also competitive with the *supervised NB* classifier. A significant difference was evident only for the three *Multi*10 subsets, i.e., only when the number of categories was relatively high.

- The poor performance of the *sKL* algorithm was due to a typical fast convergence into one big cluster which consisted of almost all documents. This tendency is due to the over sensitivity of this algorithms to “zero” probabilities in the centroid representations and it was clearly less dominant in the medium scale experiments.

4.5.6 Results for medium-scale experiments

In Table 4.5 we present the results for the medium-scale datasets. To the best of our knowledge these are the first reported results (using direct evaluation measures as precision and recall) for unsupervised methods over corpora of this magnitude (on the order of 10^4 documents).

Table 4.5: Micro-averaged precision results over the medium scale datasets. In all the unsupervised algorithms the number of clusters was taken to be identical with the number of real categories (indicated in parentheses). For *K-means, sK-means, sL1*, and *sKL* the reported precision results are the best results out of all 10 restarts. For sIB the results are for the restart which maximized $I(T_d; W)$. The *NB* classifier was trained over $1,000$ randomly chosen documents and tested over the remaining. We repeated this process 10 times and averaged the results.

$Prec(T_d)$	sIB	*sK-means*	*K-means*	*sL1*	*sKL*	*NB*
*NG*10 (10)	79.5	76.3	70.3	27.7	58.8	80.8
*NG*20 (20)	57.5	54.1	53.4	15.3	28.8	65.0
Reuters (10)	85.8	64.9	66.4	70.1	59.4	90.8
new-Reuters (10)	83.5	66.9	67.3	73.0	81.0	85.8
Average	**76.6**	**65.6**	**64.4**	**46.5**	**57.0**	**80.6**

For all the unsupervised algorithms we performed $N = 10$ different restarts and limited the number of iterations over $\mathcal{D}$ to 10. We note here that this limitation was in fact probably too low for some of the algorithms (see below). For these tests as well we applied the supervised *NB* classifier. For each test, the training set consisted of $1,000$ documents, randomly chosen out of the dataset, while the test set consisted of the remaining documents. Again, we repeated this process 10 times and averaged the results.

Note that the two Reuters datasets are multi-labeled while all our classification schemes are uni-labeled. Therefore the recall of these schemes is inherently limited in these two cases. This is especially evident for the *new-Reuters* data in which the average number of labels per document was 1.78 and hence the maximum attained (micro-averaged) recall was limited to 56%.

Our main findings are listed in the following.

- Similar to the small-scale experiments, sIB outperforms all the other unsupervised techniques, typically by a significant margin and in spite of the “unfair” comparison.
- Interestingly, sIB was almost competitive with the supervised *NB* classifier which was trained over $1,000$ labeled documents.
- Both our sequential and parallel *K-means* implementations performed surprisingly well, especially over the uni-labeled *NG*10 and *NG*20 tests. As in the small datasets, the differences between the parallel and the sequential implementation were minor.
- The convergence rate of the sIB and the *sK-means* algorithms were typically better than those of the other algorithms. In particular, sIB and *sK-means* converged for most of their initializations, while, for example, *sL1* did not converge in all restarts.

4.5.7 Improving cluster precision

In supervised text classification one is able to trade off precision with recall by defining some thresholding strategy. In the following we suggest a similar idea for the *unsupervised* scenario. Note that once a partition T_d is obtained we are able to estimate $\bar{d}(d, t_d(d))\ \forall d \in \mathcal{D}$. This measure provides an estimate of how “typical” d is in $t_d(d)$. Specifically in the context of sIB, $\bar{d}(d, t_d(d))$ is related to the minimal loss of relevant information by *not* holding d as a singleton cluster.

By sorting the documents in each cluster $t_d \in \mathcal{T}_d$ with respect to $\bar{d}(d, t_d(d))$ and “labeling” only the top $r\%$ of the documents in that cluster we can now reduce the recall while (hopefully) improving the precision. More specifically while defining the “label” for every cluster we only use documents that were sorted among

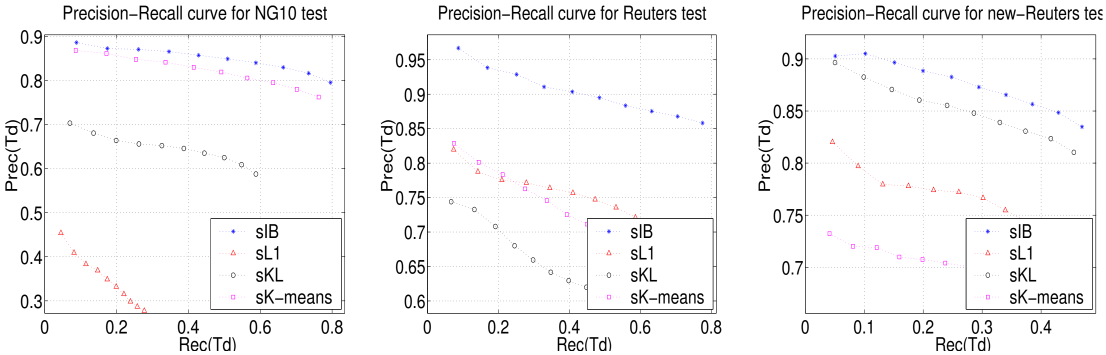

Figure 4.9: Precision-Recall curves for some of our medium scale tests. For the other tests the results were similar. Note that the results for sIB are for the specific restart which maximized $I(T_d; W)$ while for the other methods we present the best result over all 10 restarts.

the top $r\%$ for that cluster (and refer to the remaining as "unlabeled"). Note that this procedure is independent of the specific definition of $\bar{d}(d, t_d(d))$ and thus could be applied to all the sequential algorithms we tested.

In Figure 4.9 we present the Precision-Recall curves for some of our medium scale tests. Again, we find sIB to be superior to all the other unsupervised methods examined. In particular for $r = 10\%$ sIB attains very high precision in an entirely unsupervised manner for our real world corpora.

Chapter 5

Applications through Markovian Relaxation

A preliminary assumption of the IB method is that the input is given in the form of a joint distribution. Nonetheless, in many situations this may not be the most natural representation.

One important class of clustering methods deal with cases where the data are given as a matrix of pairwise distances or (dis)similarity measures. Often these distances come from empirical measurements or some complex process, and there is no direct access, or even precise definition, of the distance function. In many cases this distance does not form a metric, or it may even be non-symmetric. Such data do not necessarily come as a sample of some meaningful distribution and even the issue of generalization and sample to sample fluctuations is not well defined. Clustering algorithms that only use the pairwise distances, without explicit use of the distance measure itself, employ statistical mechanics analogies [17] or collective graph theoretical properties [34], etc. The points are then grouped based on some global criteria, such as connected components, small cuts, or minimum alignment energy. Such algorithms are sometimes computationally inefficient and in most cases it is difficult to interpret the resulting clusters; i.e., it is hard to determine a common property of all the points in one cluster - other than that the clusters "look reasonable".

A second class of clustering methods is represented by the generalized vector quantization (VQ) algorithm. Here one fits a model (e.g., Gaussian distributions) to the points in each cluster, such that an average (known) distortion between the data points and their corresponding representative is minimized. This type of algorithms may rely on theoretical frameworks, such as rate distortion theory, and provide a better interpretation for the resulting clusters. VQ type algorithms can also be more computationally efficient since they require the calculation of distances, or distortions, only between the data and the centroid models, not between every pair of data points. On the other hand, they require knowledge of the distortion function and thus make specific assumptions about the underlying structure or model of the data.

As we discussed in Section 2.2, the IB method can be used to bypass this difficulty, by introducing the concept of a relevant variable. In this case, no distortion measure need be defined in advance, and the problem amounts to optimizing the trade-off between the compression-information and the relevant information.

In this chapter, following [83], we investigate how to apply this framework in the context of pairwise clustering. We show how to define a "relevant" variable in these situations as well, which leads to an intuitive interpretation for the resulting clusters. The idea is based on turning the distance matrix into a Markov process and then examining the decay of mutual information during the relaxation of this process. The clusters emerge as quasi-stable structures during this relaxation, and are then extracted using the IB method. These clusters capture the information about the initial point of the relaxation in the most effective way. The suggested approach can cluster data with no geometric or other bias and makes no assumptions about the underlying distribution.

5.1 Pairwise distances and Markovian relaxation

The first step is to turn the pairwise distance matrix into a Markov process, through the following simple intuition. Assign a state of a Markov chain to each of the data points and define the transition probabilities between the states/points as a function of their pairwise distances. Thus the data can be considered as a directed graph with the points as nodes and the pairwise distances, which need not be symmetric or form a metric, as the lengths of the graph arcs. Distances are normally considered additive; i.e., the length of a trajectory on the graph is the sum of the arc-lengths. Probabilities, on the other hand, are multiplicative for independent events. Thus, if we want the probability of a (random) trajectory on the graph to be naturally related to its length, the transition probabilities between points should be exponential in their distance. Denoting by $d(x_i, x_j)$ the pairwise distance from x_j to x_i, then the transition probability that our Markov chain will move from point x_j at time n to point x_i at time $n+1$, is defined as,

$$p(x_i(n+1)|x_j(n)) \propto e^{(-\lambda_j d(x_i,x_j))} , \tag{5.1}$$

where λ_j is a length scaling factor defined by

$$\lambda_j = \frac{f}{\bar{d}(k,j)} , \tag{5.2}$$

where f is some constant and $\bar{d}(k,j)$ is the mean pairwise distance of the k nearest neighbors to point x_j. The details of this rescaling are not crucial for the final results, and a similar exponentiation of the distances, without our probabilistic interpretation, has been performed in other clustering works (e.g., [17, 34]). A proper normalization of each row is required to turn this matrix into a stochastic transition matrix.

Given this transition matrix, one can imagine a random walk starting at every point on the graph. Specifically, the probability distribution of the positions of a random walk, starting at x_j after n time steps, is given by the j-th row of the $n-th$ iteration of the 1-step transition matrix. Denoting by P^n the n-step transition matrix, $P^n = (P)^n$, is indeed the n-th power of the 1-step transition probability matrix. The probability of a random walk starting at x_j at time 0, to be at x_i at time n is thus:

$$p(x_i(n)|x_j(0)) = P^n_{i,j} . \tag{5.3}$$

If we assume that all the given pairwise distances are finite we obtain in this way an ergodic Markov process with a single stationary distribution, denoted here by π. This distribution is a right-eigenvector of the n-step transition matrix (for every n), since, $\pi_i = \sum_j P_{i,j}\pi_j$. It is also the limit distribution of $p(x_i(n)|x_j(0))$ for all j, i.e., $\lim_{n\to\infty} p(x_i(n)|x_j(0)) = \pi_i$. During the dynamics of the Markov process any initial state distribution will relax to this final stationary distribution and the information about the initial point of a random walk is completely lost, as described in the following.

5.2 Relaxation of the mutual information

The natural way to quantify the information loss during this relaxation process is by the mutual information between the initial point variable, $X(0) = \{x_j(0)\}$ and the point of the random walk at time n, $X(n) = \{x_i(n)\}$. That is,

$$I(n) \equiv I(X(0); X(n)) = \sum_j P_j \sum_i P^n_{i,j} \log \frac{P^n_{i,j}}{P^n_i} = \sum_j P_j D_{KL}[P^n_{i,j} \| P^n_i] , \tag{5.4}$$

where P_j is the prior probability of the states, and $P^n_i = \sum_j P^n_{i,j} P_j$ is the unconditioned probability of x_i at time n. As $n \to \infty$, all the rows $P^n_{\cdot,j}$ and the unconditional probabilities P^n_i relax to π, hence all the KL

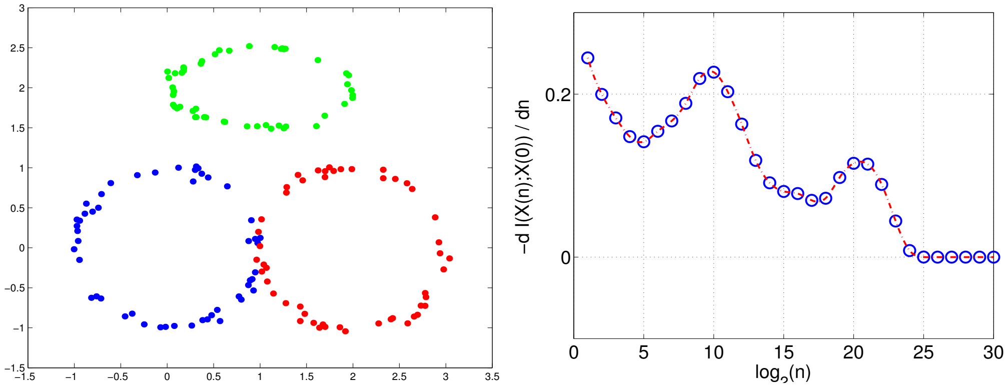

Figure 5.1: **Left:** An example of (synthetic) data, consisting of 150 points in $\mathcal{R}^2$. **Right:** The rate of information loss, $-\frac{dI(n)}{dn}$, during the relaxation (where we set $f = 1,\ k = 3$ for calculating λ_j). The information loss is slower when the "random walks" stabilize on some sub structures of the data - our proposed clusters. Thus, at these points we expect to see a local minimum in this rate. The first minimum of the rate corresponds to the emergence of the first sub-structure (the partition into three circles). The second minimum corresponds to the emergence of the second sub-structure in the hierarchy, which is the partition of the two lower circles as one cluster, versus the upper circle as the second cluster. Note that the process has no prior information about circles or ellipses.

divergences relax to zero; i.e., $I(n) \rightarrow_{n\rightarrow\infty} 0$. While it is clear that the information about the initial point, $I(n)$, decays monotonically (exponentially asymptotically) to zero, the *rate* of this decay at finite n conveys much information on the structure of the data points.

Consider, as a simple example, the planar data points shown in Figure 5.1, with $d(x_i, x_j)$ taken as the (squared) L_2 norm. As can be seen, the rate of information loss about the initial point of the random walk, $-\frac{dI(n)}{dn}$, while always positive - slows down at specific times during the relaxation. These relaxation locations indicate the formation of quasi-stable structures on the graph. Those structures form natural clusters of initial points that contain the same information on the position at time n. Another way to see this phenomenon is by observing the rows of P^n, which are the conditional distributions $p(x_i(n)|x_j(0))$. The rows that are almost indistinguishable, following the partial relaxation, correspond to points x_j with similar conditional distribution on the rest of the graph at time n. Such points should belong to the same structure, or cluster on the graph. This can be seen directly by observing the matrix P^n during the relaxation, as shown in Figure 5.2.

The quasi-stable structures on the graph, during the relaxation process, are precisely the desirable *meaningful* clusters. At these relaxation times the transition probability matrix is approximately a projection matrix (satisfying $P^2 = P$) where the almost invariant subgraphs correspond to the clusters. These approximate stationary transitions correspond to slow information loss, which can be identified by examining the derivative of the information change.

The remaining question pertains to the correct way to group the initial points into clusters. Can we replace the initial point with an initial cluster that enables prediction of the location on the graph at time n, with similar accuracy? The answer to this question is naturally provided via the IB method. In particular the compression variable is taken as $X = X(0)$, while the relevant variable is defined as $Y = X(n)$. Thus, applying one of the IB algorithms to a partially relaxed transition probability matrix will yield clusters of data points that capture the information about the position on the graph after n-steps in the most effective way.

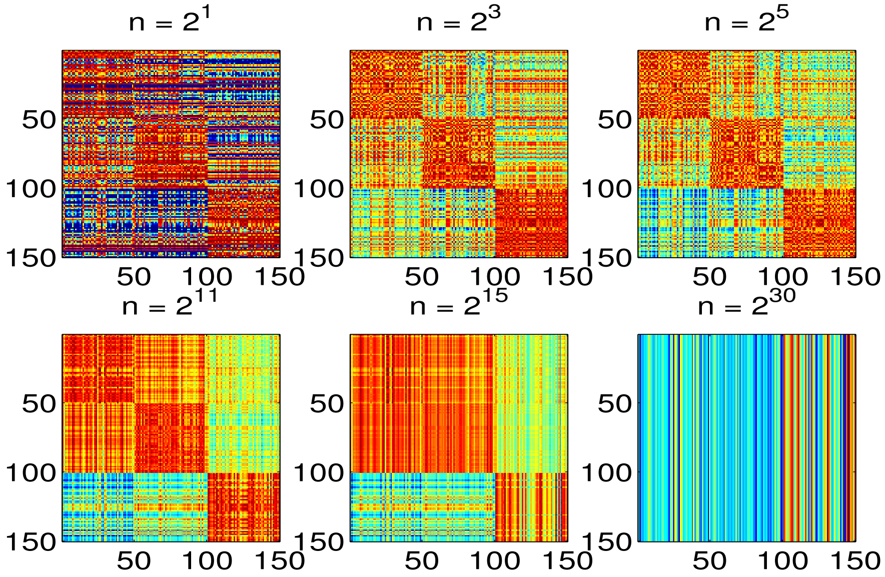

Figure 5.2: The relaxation process as seen directly on the matrix P^n, for different relaxation times, for the example data of Figure 5.1. The darker (red) colors correspond to higher probability density in every row. Since the points are ordered by the 3 ellipses, 50 in each ellipse, it is easy to see the clear emergence of 3 blocks of conditional distributions - the rows of the matrix - during the relaxation process ($n \approx 2^5$). As the relaxation continues the two upper blocks (corresponding to the two lower circles in Figure 5.1) are mixed with each other. At $n \approx 2^{15}$ there is a new quasi-stable structure, partitioning the data into two clusters. At $n \to 2^{30}$ all the rows converge to the stationary distribution of the data. At that point, all the information about the initial point is lost and we have $I(n) \to 0$.

5.3 Applications

In the following we present several applications of combining Markovian relaxation with the IB method. In all these applications, for the sake of simplicity, we used the sIB algorithm with $\beta^{-1} = 0$ (since the cardinality $|\mathcal{T}|$ already implied significant compression). We set $|\mathcal{T}|$ as the true number of classes in the data and performed ten different random initializations, from which we choose the run that maximized the relevant information.

5.3.1 The Iris data

We start with a simple application to the famous "Iris-data" [31]. These data contain geometric measures of three types of iris flowers, each represented as a 4-dimensional real vector. While one of the classes is easily linearly separable, the two others are difficult to separate in this representation.

From the raw data we calculated a 150×150 similarity matrix using the (squared) L_2 norm as the distance measure. Using Eq. (5.1) we extracted from this matrix the transition probability matrix (where we set $f = 1,\ k = 10$ for calculating λ_j). In Figure 5.3 (left panel) we present the rate of information loss during the relaxation process. Here the emergence of the three classes is harder to identify on the relaxation curve, while the separation into two clusters is easily noted. More specifically, for $n \approx 2^4$ the rate of information loss is slightly decreased, which corresponds to the first quasi-stable structure, i.e., the emergence of three clusters. Next, there is a wide range of n values (between 2^{10} and 2^{25}) that corresponds to the second quasi-stable structure, that is, the emergence of two clusters.

Although the first quasi-stable structure seems hard to identify, applying the sIB algorithm at the proper relaxation time ($n = 2^4$) reveals the original classes with only five "misclassified" points (where these points are located on the border between the corresponding classes, see the right panel of Figure 5.3). In terms of

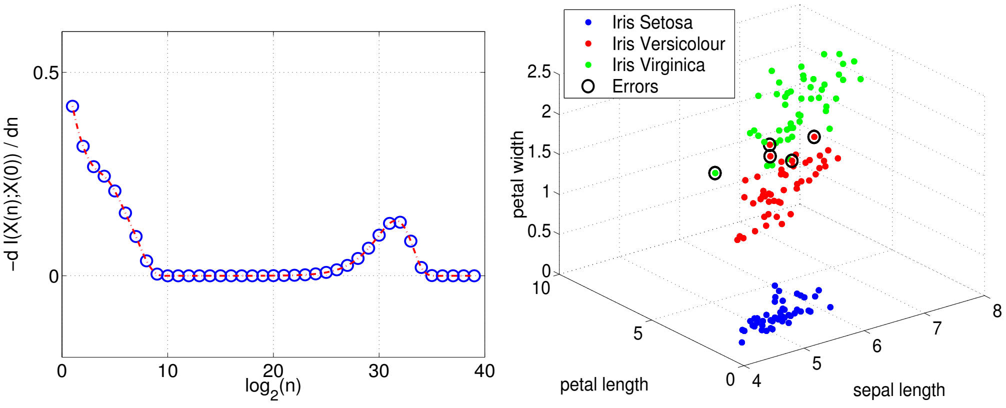

Figure 5.3: **Left:** Rate of information loss during the relaxation for the Iris data. The first quasi-stable structure emerges around $n = 2^4$. The second quasi-stable structure is present for $2^{10} \leq n \leq 2^{25}$. For $n \geq 2^{35}$ the information converges to its asymptotic zero value. **Right:** The Iris data, consisting of 150 points in $\mathcal{R}^3$ (where the second coordinate which has the lowest variance is ignored for this presentation). Applying sIB to the partially relaxed transition probability matrix (with $n = 2^4$) yields three clusters with almost perfect correlation to the true partition. The "misclassified" points are circled in the figure. Note that these points are on the border between the two corresponding classes.

information, the resulting three clusters capture $\approx 58\%$ of the original information, $I(X(0); X(n)) \approx 1.65$.

5.3.2 Gene expression data analysis

A more interesting application was obtained on well known gene expression data, the *Colon cancer* dataset provided by Alon *et. al* [1]. These dataset consists of 62 tissue samples out of which 22 came from tumors and the rest are "normal" biopsies of colon parts from the same patients. Gene expression levels were given for $2,000$ genes (oligonucleotides), resulting in a 62 over $2,000$ matrix.

As done in other studies of these data, the pairwise distances we calculated were based on the *Pearson correlation*, $K_p(u, v)$ (see, e.g., [28]) between the u and v expression rows. Specifically,

$$K_p(u, v) \equiv \frac{E\left[(u_i - E\left[u\right])(v_i - E\left[v\right])\right]}{\sqrt{\mathrm{Var}\left[u\right] \mathrm{Var}\left[v\right]}} \, . \tag{5.5}$$

We transformed these similarity measures into "distances" through a simple transformation, given by $d(u, v) = \frac{1-K_p(u,v)}{1+K_p(u,v)}$. Using these distances we obtained the transition probabilities through Eq. (5.1), where we set $f = 5,\ k = 5$ for calculating λ_j. Note that tuning these two parameters is done based on the raw data alone, and in particular without any use of the true labels, where all we need to look for is "good" behavior in the rate of information loss, indicating the emergence of quasi-stable structures.

In the left panel of Figure 5.4 we present the rate of information loss for these data. For $n > 2^{25}$ this rate clearly starts to decrease, indicating the emergence of some sub-structure in the data. We further applied the sIB algorithm to all the partially relaxed matrices (for $n = 2^0 \ldots 2^{45}$), and measured the correlation of the resulting two clusters with respect to the true partition (i.e., the micro-averaged precision, as defined in Section 4.5.2). In the right panel of Figure 5.4 we present these results. As expected, when the sub-structure starts to emerge ($n \approx 2^{26}$), the sIB algorithm recovers the original tissue classes with very high accuracy. In particular, only 8 samples are "misclassified". For comparison, seven sophisticated *supervised* techniques were applied in [9] to these data. Six of them had at least 12 misclassified points , and their best results had 7 missclasifed tissues. As the information converges to its asymptotic zero value, the accuracy also drops to its baseline value that corresponds to a random partitioning of the samples into two clusters.

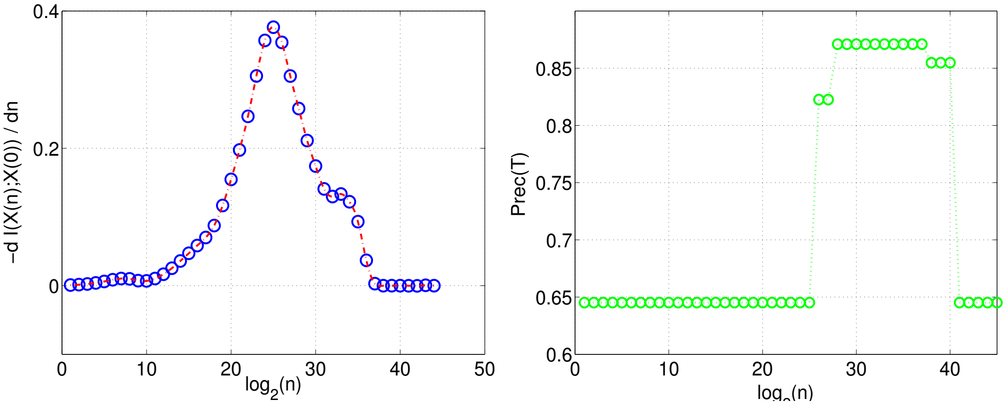

Figure 5.4: **Left:** The rate of information loss for the *colon cancer* data. The rate starts to decrease for $n \approx 2^{26}$, indicating the emergence of a sub-structure in the data. However, for $n \approx 2^{40}$, in spite of the slower rate, all the information is already lost since all the conditional distributions converge to the stationary distribution. **Right:** Correlation (or micro-averaged precision) of the two clusters extracted by sIB with respect to the true labels, for different relaxation times. Note that as the rate of information loss is decreasing, the accuracy of the extracted clusters is increasing. That is, the structure of the data becomes evident and easy to recover for these relaxation times.

5.3.3 Unsupervised OCR

Last, we consider applying our methodology to a standard Optical Character Recognition (OCR) task. The *MNIST* database [1], consists of $60,000$ (training) examples of handwritten digit images. Each example is associated with a digit from '0' to '9', and represented as a 28×28 pixel image, where each pixel can have a gray level between 0 and 255.

Typically these data are used for evaluating *supervised* classification techniques (e.g., [8]). However, since we were interested in an *unsupervised* application, we used only a small subset of these data. Specifically we randomly chose 300 examples, evenly distributed among the digits '1', '3' and '8'. To extract the transition probabilities we used the *Pearson correlation* again, exactly as in the previous section (where we set $f = 15,\ k = 2$).

In Figure 5.5 we present the rate of information loss. Note that there are two minima present in this curve. The first, at $n \approx 2^{27}$, corresponds to the emergence of the first sub-structure, namely the partition into three classes. Applying the sIB algorithm (with $|\mathcal{T}| = 3$) to this partially relaxed matrix, yields three clusters which are well correlated with these classes (the micro-averaged precision is 91.3%). Moreover, with only three clusters we have $I(T; X(n)) = 0.89$, which is $\approx 45\%$ of the original information.

The second minimum occurs around $n \approx 2^{34}$. At this point, a new sub-structure emerges that naturally corresponds to partitioning the data into the '1' digits versus all the rest. That is, at this point the classes '3' and '8' are mixed, hence we are at a lower level of the hierarchy of structures for these data. Applying the sIB algorithm (with $|\mathcal{T}| = 2$) to this matrix, yields two clusters with almost perfect correlation to this dichotomy (the micro-averaged precision is 96.7%). In terms of information, $I(T; X(n)) \approx 0.43$ which is 49% of the original information.

5.4 Isotropic blurring versus relaxation

In the IB method, when varying the trade-off parameter β (the inverse "temperature" of the system), one explores the structure of the data in various resolutions. For high β values, the resolution is high and each point eventually appears in a cluster of its own. For low β all points are grouped into one cluster. This

[1] Available at *http://www.research.att.com/˜yann/exdb/mnist/index.html*.

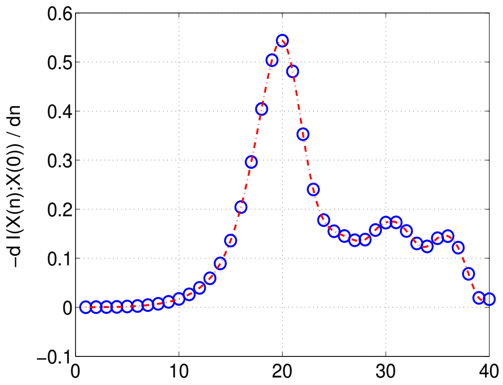

Figure 5.5: Rate of information loss for a subset of the *MNIST* data. The rate is first minimized for $n \approx 2^{27}$. Applying the sIB algorithm at this point yields three clusters which are well correlated with the true partition into '1', '3', and '8' digits. The second minimum for the rate is around $n \approx 2^{34}$. At this point classes '3' and '8' are already mixed, hence we have a new sub-structure. Applying the sIB at this point extracts two clusters that almost perfectly match the partition of '1' digits versus '3' and '8' digits.

process resembles the appearance of the structure during the relaxation. However, there is an important difference between these two mechanisms.

In the IB case clusters are formed by *isotropically blurring* the conditional distributions that correspond to each data point. Points are clustered together when these distributions become sufficiently similar. This process is not sensitive to the global topology of the graph representing the data. This can be understood by looking at the example in Figure 5.1. If we consider two diametrically opposed points on one of the ellipses, they will be clustered together only when their blurred distributions overlap. In this example, unfortunately, this happens when the three ellipses are completely indistinguishable. A direct application of some IB algorithm to the original transition matrix is therefore bound to fail in this case.

In the relaxation process, on the other hand, the distributions are merged through the Markovian dynamics on the graph. In our specific example, two opposing points become similar when they reach the other states with similar probabilities following partial relaxation. This process better preserves the fine structure of the underlying graph, and thus enables finer partitioning of the data.

It is thus necessary to combine the two procedures. In the first stage, one should relax the Markov process to a quasi-stable point in terms of the rate of information loss. At this point some natural underlying structure emerges, and reflected in the partially relaxed transition matrix, P^n. In the second stage we use one of the IB algorithms to identify the information preserving clusters. As shown in the previous sections, this combination enables to successfully extract a hierarchy of structures out of pairwise distance data.

Chapter 6

Discussion and Further Work

In the first part of this thesis we introduced the single-sided IB principle. From a theoretical perspective we showed that it can be considered as an extension to rate distortion theory. In both cases the underlying principle is constraint minimization of the compression-information between the source random variable, X, and its new representation, T. However, while in rate distortion theory the constraint is over the expected distortion, in the IB case it is associated with the minimal level of relevant information about the target variable, Y. Consequently, the problem setup is completed once the joint distribution, $p(x, y)$ is provided. No distortion measure need be defined in advance, and the input statistics are fully characterized by a single function, the relevance-compression function.

As mentioned previously, a dual formulation of the IB principle is to *maximize* the relevant information under a constraint over the maximal level of the compression-information. Taking this view, the IB principle is related to channel coding theory, in which the fundamental problem is maximizing the information transmitted through a channel, under a constraint over the channel properties.

Interestingly, this type of duality was already pointed out by Shannon himself. In his words: *"There is a curious and provacative duality between the properties of a source with a distortion measure and those of a channel. This duality is enhanced if we consider channels in which there is a "cost" associated with the different input letters and it is designed to find the capacity subject to the constraint that the expected cost not exceed a certain quantity (...) . This problem amounts, mathematically, to* maixmizing *a mutual information (...) with a linear inequality as constraint. The solution of this problem leads to a capacity cost function $C(a)$ for the channel. It can be shown readily that this function is* concave *downward (...) . In a somewhat dual way, evaluating the rate distortion function $R(D)$ for source amounts, mathematically, to* minimizing *a mutual information under variation (...), again with a linear inequality as constraint. The solution leads to a function $R(D)$ which is convex downward."* [1]

While in this classic formulation the duality requires associating a "cost" with the channel input letters (which provides the analogous component to the distortion constraint), in the IB formulation the duality is, in this sense, articulated in the principle in a more natural way. Since in the "rate distortion view" of the IB no distortion is pre-defined, there is no need to refer to a "cost" while taking the "channel coding" perspective. Both sides of the IB principle are constructed out of exactly the same concept of mutual information. We wish to minimize the compression-information while maximizing the relevant information, and apparently it is not important which one we wish to use as a constraint and which one we choose to optimize.

Moreover, in contrast to standard rate distortion thoery, the constraint in the IB principle is *not* linear in the paramteres of the problem, the mapping $p(t \mid x)$. Nevertheless, Tishby *et al.* [82] succeeded in showing that it is possible to characterize the form of the optimal solution, even in this more complicated scenario (see Section 2.4). Additionally, while in rate-distortion the problem is defined with respect to a fixed set

[1] This quote from Shannon is taken from a recent position paper by S. K. Mitter, in *IEEE Information Theory Society Newsletter*, December 2000. The discussion in [15] is also insightful in this context.

of representatives, in the IB case there is no such prerequisite. On the contrary, the representatives (given as cluster centroids, $p(y \mid t)$ in this case) depend directly on the mapping $p(t \mid x)$, and thus necessarily change while this mapping is optimized. Hence, the general formulation of the problem is defined with respect to any choice of representatives, and so is the unique characteristic function of $p(x, y)$, the relevance-compression function.

This type of joint optimization typically comes with a cost, and in this context the IB problem is no exception. Specifically, the IB-functional is not convex in all of its arguments simultaneously. As a result, constructing optimal solutions in practice can be shown to be NP-hard in general (see, e.g., [33]). Hence, different heuristics must be employed. We have presented several such heuristics and demonstrated their applicability in different contexts. In particular we saw that in many cases it is possible to extract (sometimes extremely) compact representations that still maintain a significant fraction of the relevant information about the target variable.

These applications also demonstrate the practical implications of the IB method. For example, our results for document clustering are superior to other state-of-the-art clustering techniques. This superiority is not only in terms of the IB-functional but also in terms of a "practically oriented" measure; namely, how well the extracted clusters correlate with the corpus topics. Moreover, in [74] we show that our results are even superior to algorithms that are especially designed for text classification tasks. In Appendix B we provide a theoretical analysis that shed more light on these empirical findings.

We further demonstrated that the IB method can be applied to analyze a complex real world data given in the form of a natural language text. This analysis can take different forms. One may extract word clusters that preserve information about the topics in the corpus (Section 4.1, Section 4.3 and [78, 87]), the origins of the documents (Section 4.1), the documents themselves [77], the neighboring words [41, 60], and so on. Additionally, the method is certainly not limited to text applications, and in principle can be applied to any type of (co-occurrence) data that can be represented as a joint distribution. A variety of such applications to different data types are presented in [38, 56, 68, 75, 83] and in Chapter 5. All these presumably different tasks are addressed in a well defined way through a single information theoretic principle, the IB principle. Moreover, the interpretation of the results is objective and data-independent. The quality of the clusters is quantified explicitly in terms of the trade-off between the compression information and the relevant information that these clusters capture.

An important issue is to characterize the relationship between the IB method and other probabilistic clustering techniques. In particular, a standard and well established approach to clustering is Maximum likelihood (ML) of mixture models (see, e.g., [53]). Although both approaches stem from conceptually different motivations, it turns out that in some cases there are some mathematical equivalences between them. As a result, it is possible to show that under certain conditions, every algorithm that solves one of the problems induces a solution to the other. These results are discussed in detail in Appendix A.

An interesting special case of the IB framework relates it to the notion of a time series, and in particular to extracting compact representations of the past of a series that maximally preserve the information about its future. This type of application was already mentioned in [11]. Furthermore, Bialek *et al.* [10] formulated the notion of *predictive information* in a stream of data $x(t)$. Denoting the last k symbols of the stream (or series) as $x_{past} \equiv [x(-k)\ x(-k+1)\ \ldots\ x(-1)] \in \mathcal{X}^k$ and the next k' symbols as $x_{future} \equiv [x(0)\ x(1)\ \ldots\ x(k'-1)] \in \mathcal{X}^{k'}$, the predictive information is defined as the mutual information between x_{past} and x_{future}, i.e., $I(k; k') \equiv \sum_{x_{past}, x_{future}} p(x_{past}, x_{future}) \log \frac{p(x_{future} | x_{past})}{p(x_{future})}$. A thorough analysis in [10] relates this definition to the level of complexity of the series $x(t)$. In our context, a natural question to ask is whether it is possible to compress x_{past} while still preserving most of the information about x_{future}. Intuitively it is clear that not all of the details in x_{past} are indeed informative about x_{future}, therefore this type of application seems highly suitable for our framework. Although we did not examine it directly, a somewhat reminiscent application is presented in Section 11.3.1.

Chechik and Tishby [19] have recently pointed out that in many cases it is also possible to specify what is *irrelevant* for the task at hand. Identifying the relevant structures in the data can thus be improved by also *minimizing* the information about another, irrelevant variable. One way to formalize this notion is to add an "irrelevant-information" term to the IB-functional given in Eq. (2.13). Specifically, Chechik and Tishby considered the functional $\mathcal{L} = I(T;X) - \beta\left[I(T;Y^+) - \gamma I(T;Y^-)\right]$, where Y^+ and Y^- denotes the relevant and the irrelevant variables, respectively, and γ is a second Lagrange multiplier that determines the trade-off between preservation of information about Y^+ and loss of information about Y^-. Using again the same IB Markovian relation (Eq. (2.10)), it is easy to verify that the form of the optimal solution is analogous to the solution of the original IB principle (Eq. (2.16)). The only difference is in the form of the exponent. In addition to the "relevant-distortion" term, $D_{KL}[p(y^+ \mid x)\|p(y^+ \mid t)]$, there is also an "irrelevant-distortion" term, $D_{KL}[p(y^- \mid x)\|p(y^- \mid t)]$ multiplied by γ and with an opposite sign [19]. While the original IB principle is related to rate distortion theory, this new extension of it is related to rate distortion with side information (see [20], page 438).

Last, another contribution, which is closely related (and somewhat complementary) to the IB method, was recently introduced by Globerson and Tishby [37]. In contrast to the clustering based approach discussed in this thesis, they suggested extracting *continuous feature functions* of X that maximize the information about Y, under some natural constraints that these functions should maintain. More precisely, denoting these functions by $\vec{\phi} \equiv \{\phi_1, \ldots, \phi_d\}$, $\phi_i(x) : X \to \mathcal{R}$, given $p(x,y)$ the problem is to find $\vec{\phi}^* = \mathrm{argmax}_{\vec{\phi}(x)} \min_{\tilde{p}(x,y)\in Q} I[\tilde{p}(x,y)]$, where Q is the class of all distributions $\tilde{p}(x,y)$ with marginals $p(x),\ p(y)$ and *expected measurements* $\langle\vec{\phi}(x)\rangle_{\tilde{p}(x|y)} = \langle\vec{\phi}(x)\rangle_{p(x|y)},\ \forall y \in \mathcal{Y}$. Thus, as in the original IB formulation, one faces a *min-max* problem of minimizing information on the one hand and maximizing it on the other. Note, though, that while in the IB case the minimization part was required to force compression (as in rate distortion), here the compression is implied by the choice of the input parameter d which determines the dimension of the extracted new representation. The minimization of $I[\tilde{p}(x,y)]$ in this case is similar in motivation to the maximum entropy principle [46], where it guarantees that the representation $\tilde{p}(x,y)$ will contain *only* the information given by the measurement values $\vec{\phi}(x)$.

As shown by Globerson and Tishby, although this variational principle does not define a generative statistical model for the data, the resulting distribution $\tilde{p}(x,y)$ is necessarily of an exponential form and can be interpreted as a generative model in this class. Hence, the above problem is equivalent to a (maximum likelihood) problem of minimizing the KL divergence between the input distribution $p(x,y)$ and a family of distributions of an exponential form. Specifically, the approximation is given by $\tilde{p}(x,y) \propto e^{\sum_{i=1}^{d}\phi_i(x)\psi_i(y)}$, where the functions $\psi_i : Y \to \mathcal{R}$ provide simultaneously continuous functions of Y which are informative about X. In a sense, the two sets of d functions, $\vec{\phi}$ and $\vec{\psi}$, can be considered as *approximate* sufficient statistics for a sample of one variable about the other one. Due to its tight link to the concept of sufficient statistics (see, e.g, [20], page 36) this method was termed Sufficient Dimensionality Reduction (SDR).

Clearly, both approaches stem from similar motivations and seek similar goals of extracting a low dimensional, yet informative representation of a given joint distribution $p(x,y)$. Nonetheless, clarifying the relationships between these two approaches still requires further investigation. In some cases, where the relation between X and Y comes from some hidden low-dimensional *continuous* structure, applying SDR seems more reasonable. On the other hand, if the distributions $p(y \mid x)$ utilize the whole simplex in $\mathcal{R}^{|\mathcal{Y}|}$, but still form some natural clusters, applying SDR will typically fail to yield significant results, while an IB clustering approach may reveal the hidden structure. Hence, for different input distributions one approach might be more appropriate than the other. An interesting issue is to try to combine these two approaches. That is, in some cases it might be useful to apply SDR for extracting a low dimensional representation of the original data. This (presumably more robust) representation could then be used as the input for an IB algorithm to further compress the data without losing much of the relevant information. The opposite scheme where we start with IB clustering and continue through SDR might also be plausible.

6.1 Finite sample effects

The underlying assumption of our methodology is that we have access to the true joint distribution, $p(x, y)$. However, in practice, all we have are empirical estimates based on a finite sample of this distribution. Although it might be possible to characterize the required sample complexity (which depends on $|\mathcal{X}|$ and $|\mathcal{Y}|$) for a *uniform* convergence of these estimates to their true values, an intriguing question is how sensitive our framework is to finite sample effects.

Our empirical findings show that in practice we can achieve reasonably good performance while only using estimates of $p(x, y)$. For example, in Section 4.4 we saw that the sIB algorithm can achieve good results, even for very small sample sizes. Specifically in this case we had $|\mathcal{X}| = 500,\ |\mathcal{Y}| = 2,000$ (i.e., 10^6 entries in the joint distribution $p(x, y)$) and for a sample size of $N = 50,000$ the obtained clusters were typically highly correlated with the "true" partitioning of the data. In Section 4.5 we saw that in a real-world application, our results are superior to other state-of-the-art techniques.

Leaving practical considerations aside, a remaining question is the theoretical interpretation of our results when the true $p(x, y)$ is not available. It turns out that it is possible to suggest an alternative view of our framework which stems from a long line of works in the statistical literature. Specifically we are interested in different methodologies that were suggested over the years regarding the problem of when and how to collapse (i.e., to merge) rows or columns in a given two-way contingency table. The motivation behind these works is 'to get a more parsimonious and compact description of the data' while revealing existing 'patterns of association'. Typically, a statistical criterion (e.g., the reduction in the chi-square statistic) is used to decide which rows (or columns) should be merged (see [36] and the references therein). To interpret our work in this context, recall that the JS measure is closely related to a classic statistical test, the two-sample problem (Section 1.2.5). However, the same divergence measure is the cornerstone of the merging criterion used by the aIB and the sIB algorithms (Eq. (3.8), where for simplicity we assume here $\beta^{-1} = 0$). Thus, we may consider the input distributions $p(y \mid x)$ as finite sample estimates of some hidden statistical sources. In this view, while using the IB merging criterion we are in fact seeking for the pair of (empirical) distributions which are most likely to be considered as two (types of) samples from the same statistical source. Once such a pair is found we merge it to a single entity, and repeat the process until some halt criterion is satisfied. Thus, although our original motivation is different, at least some of the IB algorithms are well motivated under this framework, with no need to require access to the true distribution, $p(x, y)$.

Another issue which is affected by finite sample effects is the estimation of the relevance-compression function. In a limited sampling scenario, direct measurements of the information terms involved in our analysis (based on the empirical distribution) will generally be incorrect, and in fact (on the average) over estimated [84]. Different methods have suggested how to correct this upper bias. One simple and intuitive approach is described in [57]. In this approach one randomly *shuffles* all the entries of the empirical joint distribution and calculates the "information" in the resulting random matrix. Repeating this procedure for several independent trials and averaging the results typically yields a reasonable estimate of the correction term. Hence, in principle, while estimating the relevance-compression function, it is possible to use such an approach to correct our estimates.

6.2 Future research

The analysis and the results presented in the first part of this thesis raise several issues that call for further investigation. We now briefly consider a few such examples.

6.2.1 A relevant-coding theorem?

The definition of the relevance-compression function (Definition 2.3.1), $\hat{R}(\hat{D})$ is essentially a mathematical definition. In this sense it is analogous to the mathematical definition of the rate-distortion function, $R(D)$ (Eq. (2.2)). However, rate distortion theory provides an alternative definition to this function, which is sometimes referred to as an "operational" definition. This definition is based on the concept of a rate-distortion *code* and its associated distortion (see [20], page 340). Specifically, the rate-distortion function is then defined as the infimum of rates R such that there *exists* a (possibly infinite) sequence of rate-distortion codes with an associated distortion which is asymptotically upper bounded by the distortion constraint, D. Note that this definition does not directly involve the concept of mutual information.

The first main result of rate distortion theory shows that these two definitions are equivalent. In particular, it is shown (e.g., [20], page 351) that the rate-distortion function is achievable; in other words, that for any D and any $R > R(D)$ there exists a sequence of rate-distortion codes with rate R and asymptotic distortion D. In this sense it means that the bound defined by the rate-distortion function is tight. Specifically, by increasing the length of the transmitted blocks, in principle one can always achieve the minimal rate defined by this function without exceeding the distortion constraint.

A natural goal is to try to formalize the IB analysis in a similar way. In particular, such an analysis will require a rigorous definition of a "relevant code" associated with a relevant-distortion term, which are both based solely on the input distribution, $p(x, y)$. These definitions should further lead to an "operational" definition of the relevance-compression function that does not directly involve the compression-information, $I(T; X)$. The next step would be to try to extend the above mentioned rate distortion theorem to our context. Specifically, we should try to verify whether this (potential) definition is equivalent to our original mathematical one. Lastly, one should search for an (asymptotic) existence theorem, showing that such "relevant codes" which satisfy the relevant information constraint, while utilizing a minimal rate (as defined by $\hat{R}(\hat{D})$) do exist. Clearly, this issue calls for a separate investigation which is beyond the scope of this work.

6.2.2 Model selection and avoiding over-fit

A challenging question in cluster analysis is the estimation of the "correct" number of clusters in the given data. As discussed in Section 2.3, in our context the number of clusters, $|\mathcal{T}|$ is related to the trade-off parameter β. Low β values imply significant compression, which in turn suggests a relatively small number of clusters. In contrast, high β values shift the focus toward the relevant information term, by that suggesting that a large number of clusters should be employed.

Hence, the question of setting the "correct" number of clusters can be (roughly) translated into the question of setting the appropriate β value. One approach to handle this issue, already suggested in [60], is to apply *generalization* considerations. More precisely, Pereira *et al.* suggested splitting the input data into a training set and held out data (i.e., test set). Using the dIB algorithm, the training data are then clustered for monotonically increasing β values. For each such value, the expected relevant-distortion (Section 3.1.1) is given by $\langle D \rangle \equiv \sum_{x,t} p(x)p(t \mid x) D_{KL}[p(y \mid x) \| p(y \mid t)]$, where all the distributions are estimated based on the training data alone. As β increases, additional clusters are employed (through phase-transitions, or cluster bifurcations), and $\langle D \rangle$ monotonically decreases. To determine a good stopping point for this process, it was suggested to consider a "generalization" expected relevant-distortion term, defined by $\langle D_h \rangle \equiv \sum_{x,t} p(x)p(t \mid x) D_{KL}[p_h(y \mid x) \| p(y \mid t)]$, where $p_h(y \mid x)$ are estimated from the held-out data. This term will initially also decrease as β increases, but at some critical β value, it is expected to change its tendency, i.e., to start increasing [60]. Roughly speaking, this (empirical) phenomenon indicates that from this point on we are over-fitting our training data, and consequently losing the generalization power of our clusters. Repeating this process for different splits into training and test set will presumably yield an estimate of this critical β value.

A more rigorous approach can be achieved by applying statistical learning theoretical techniques (see, e.g., [86]), which we briefly discuss in the following. First, recall that our fundamental quantity is $d(x,t) \equiv D_{KL}[p(y \mid x) \| p(y \mid t)]$, which governs the form of the optimal solution to the IB-functional (Eq. (2.16)). However, in practice we have a finite sample estimate, given by $\hat{d}(x,t) \equiv D_{KL}[\hat{p}(y \mid x) \| p(y \mid t)]$, where $\hat{p}(y \mid x)$ is estimated using our input data. Our aim is to provide a bound to the gap between these two values. Taking the (strong) assumption that our finite sample estimates of $p(y \mid x)$ converge uniformly to their true values, a general form of such a bound is given by $Pr\{\|d(x,t) - \hat{d}(x,t)\| > \varepsilon\} < \delta$, where δ is some (small) safety parameter. The gap is then (probabilistically) bounded by ε, which in general depends on the sample size N and on the *complexity* of the family of distributions $p(y \mid x)$, which we roughly denote here through α. In principle, such a bound typically implies that with probability $(1-\delta)$, we have $\hat{d}(x,t) \approx d(x,t) \pm f(\alpha)N^{-\frac{1}{2}}$, where $f(\alpha)$ is some function of α. This can be transformed into $e^{-\beta \hat{d}(x,t)} \approx e^{-\beta d(x,t)} e^{\pm \beta f(\alpha) N^{-\frac{1}{2}}}$. We may refer to the right-hand side as a multiplication of a "signal" (given by the first term) and "noise" (the second term). If we further assume that the "signal" is approximately of some known constant value, $\bar{d} \approx \beta \cdot d(x,t)$, we may argue that the left hand side cannot be trusted beyond some critical β value, given by $\beta_c \approx \bar{d} \cdot \frac{\sqrt{N}}{f(\alpha)}$ (since at that point the "signal" exponent and the "noise" exponent are of the same magnitude). Note that an interesting possibility is to estimate β_c empirically, using the previous mentioned approach of held-out data. Once this value is estimated, we can in principle obtain an *empirical* estimation of $f(\alpha)$ (through our last equality), which provides an estimate about the complexity of the family of distributions, $p(y \mid x)$.

Finally, we note here that a simple, yet plausible approach is to use our estimates of the relevance-compression function. A common empirical finding in general clustering applications is that the averaged distortion between data objects and cluster centroids decreases monotonically as the number of clusters is increased, but at some point this decrease flattens markedly (see, e.g., [40]). It is therefore intuitively reasonable to use the location of such an "elbow" as an indication of the "appropriate" number of clusters. Applying this idea in our context simply means to look for sudden drops in the estimated relevance-compression curve (which more rigorously might be characterized through the second derivative of this curve). It is important to keep in mind, though, that the question of how many clusters to use might have more than one answer. In particular, if there is some natural *hierarchical* structure in the input data, different numbers of clusters will correspond to different levels in this hierarchy, and each of these solutions should be considered. In principle, identifying these different resolutions can be done by considering the rate of the increase in β along the relevance-compression curve. A detailed discussion of this issue will be presented elsewhere.

6.2.3 Bounding the gap from the relevance-compression function

As discussed in Section 2.3, it is possible to consider the quality of the obtained clusters in the normalized relevance-compression plane. In particular, there are natural upper bounds over $I(T;Y)$ and $I(T;X)$, given by $I(X;Y)$ and $H(X)$, respectively. However, a tighter upper bound is defined through the relevance-compression function, $\hat{R}(\hat{D})$. Given some minimal required level of relevant information, this function characterizes precisely the minimal achievable level of compression. Unfortunately, while we can estimate $I(X;Y)$ and $H(X)$, estimating this function is not simple. Nonetheless, attaining reasonable bounds to this function is of significant practical importance. For example, let us assume that applying one heuristic extracts clusters that maintain 50% of the original information, $I(X;Y)$, while the compression-information term is 20% of its original value, $H(X)$. Since any heuristic we apply can in general guarantee only *locally* optimal solutions, it is certainly not clear in this situation how far we are from the *global* optimum. In other words, should we apply other heuristics or perhaps be satisfied with the current solution?

Providing some non-trivial bounds to the relevance-compression function can therefore guide us to a useful answer. Note that providing such bounds does not necessarily require constructing better clustering solutions, but rather suggesting more precise estimates as to the quality of the current solution. The fact that the relevance-compression curve is concave and that its slope is given by β^{-1} (Proposition 2.3.2) can provide some guidance. In particular this means that even if we can bound only a small number of discrete points on this curve (for known β values), it might lead to a reasonable bound over the whole curve (using simple geometrical considerations). However, even bounding a single point on this curve is in general a difficult task.

A possible approach to address this issue is through spectral analysis techniques. For example, it is possible to relate the β values for which cluster bifurcations emerge to the singular values of the Covariance matrix, corresponding to the distortions between the $p(y \mid x)$ distributions and the centroids, $p(y \mid t)$ in the current solution (see [63]). Alternatively, we may consider the stochastic matrix $P_{i,j} \propto e^{-\lambda \cdot d_{i,j}}$ (see Eq. (5.1)) where $d_{i,j}$ is, e.g., the KL divergence between $p(y \mid x_i)$ and $p(y \mid x_j)$. It is intuitively clear that the singular values of this matrix are closely related to the form of the relevance-compression function. To demonstrate this we consider two extreme, yet informative scenarios. First we assume that $p(y \mid x)$ (and hence, $p(x, y)$) is deterministic and in particular diagonal, which in fact implies $X \equiv Y$. In this case, any attempt to compress X will obviously lose a significant fraction of "relevant" information, therefore the normalized relevance-compression function is necessarily very close to the main diagonal in the normalized relevance-compression plane (much like the lower curve in the right panel of Figure 2.6). It is easy to verify that in this case $\{P\}_{i,j}$ is also diagonal, hence it is a full rank matrix and all its singular values equal one. In other words, a situation of "bad" relevance-compression curve is translated into constant singular values of $\{P\}_{i,j}$. On the other extreme, let us assume that $p(x, y)$ consists of k blocks, where in each block all the $p(y \mid x)$ distributions are equal to each other. Assuming that $k \ll |\mathcal{X}|$ it is clearly possible to construct a compact representation of k clusters, without losing any relevant information, which means that a "good" relevance-compression curve (as in the upper curve of the right panel of Figure 2.6) exists for these data. In this situation, it is easy to verify that $\{P\}_{i,j}$ will be k-blocks diagonal. That is, its first k singular values will be constant, while all the rest equal zero. Hence, a "good" relevance-compression curve is translated into a situation where only a small number of the singular values of $\{P\}_{i,j}$ are positive, while all the rest approach zero. Relaxing these two extreme examples, it is possible to construct more realistic scenarios, where the form of the singular values of $\{P\}_{i,j}$ determines the form of the relevance-compression curve. However, a more rigorous analysis is required.

Lastly, we note that in rate distortion theory there are special cases corresponding to specific assumptions about the input data for which an analytic closed-form expression can be obtained to the rate-distortion function, $R(D)$ (see, e.g., [20], page 342). Thus, it is reasonable to expect that such cases also exist in our context. Characterizing these situations along with their corresponding relevance-compression functions is left for future research.

6.2.4 Dealing with continuous distributions

A simplifying assumption, taken throughout this thesis, is that the input random variables, X and Y are both discrete, and so is the constructed compression variable. A natural direction for future research is to extend our analysis into the context of continuous random variables. Partial extensions, where, e.g., X is still discrete but Y is continuous, should also be of interest. It seems that much of the mathematical derivation presented in Chapter 2 should hold in this case as well. Moreover, for special cases this analysis might be simplified. For example, if all the representative distributions, $p(y \mid x)$ are given in the form of Gaussians, it should be reasonable to constraint the centroids $p(y \mid t)$ to the form of Gaussians mixtures. Additionally, in this case there are sufficient statistics for the representative distributions, which might also be exploited in the analysis.

A more general approach for adapting the IB framework to handle the compression of continuous distributions, given in the form of some parametric family, might be obtained through the multivariate IB method. This recent extension of the single-sided IB framework is the topic of next part of this thesis.

Part III

Multivariate Information Bottleneck

Chapter 7

Introduction

The original formulation of the single-sided IB principle concentrated on compressing one variable, X, while preserving the information it maintains about some other, relevant, variable Y. This formulation is inherently a-symmetric. Only X is compressed while only Y serves as a relevant variable. A more symmetric formulation would ask for two systems of clusters: one of X and one of Y that are informative about each other. A possible application is relating documents to words, where we seek clustering of documents according to word usage, and a corresponding clustering of words. Clearly, the two systems of clusters are in interaction, and we want a unifying principle that shows how to construct them simultaneously.

Another possible extension of the original IB formulation is to compress X into several independent systems of clusters. Our aim here is to capture independent aspects of the information X conveys about Y. A possible example is the analysis of gene expression data, where multiple independent distinctions about tissues (healthy vs. tumor, epithelial vs. muscle, etc.) are relevant for the expression of genes.

Furthermore, it is possible to think of more complicated scenarios, where there are more than two input variables. A most general formulation would require considering multiple compression variables that are inter-related by compressing different subsets of the input variables, while maximizing the information about other pre-defined subsets. In this part we provide such a principled general formulation.

To address this issue, we first need to define the amount of information that the variables $X_1, \ldots, X_n,\ n > 2$ contain about each other. To that end we use the concept of *multi-information*, which is a natural extension of the concept of mutual information we used earlier. Our approach further utilizes the theory of probabilistic graphical models such as *Bayesian Networks* for specifying the systems of clusters and which information terms should be maintained. These concepts and their relationships are discussed in the next section.

In Chapter 8 we present the multivariate IB principle. In particular, we use one Bayesian network, denoted as G_{in}, to specify a set of variables which are compressed versions of the observed variables (each new variable compresses its parents in the network). A second network, G_{out}, specifies the relations, or dependencies, that should be maintained or predicted (each variable is predicted by its parents in the network). We formulate the general principle as a trade-off between the multi-information each network carries. We want to minimize the information maintained by G_{in} and at the same time to maximize the information maintained by G_{out}. We further give another interpretation to this principle, as a trade-off between compression of the source (given by G_{in}) and fitness to a *target model*, where the model is described by G_{out}. We discuss the relations between these two formulations in Section 8.3.

We show that, as with the original IB, it is possible to characterize the form of the optimal solution to the general multivariate principle. This derivation, including some concrete examples, is given in Chapter 9. In Chapter 10 we further show that all the four algorithmic approaches for the original IB-problem are naturally extended into the multivariate case, which enables one to construct solutions in practice.

There are many possible applications to this new principle and algorithms. In Chapter 11 we consider just a few of them. In particular, we apply the method to several real world problems over a variety of data types, including text processing applications, gene expression data analysis, and protein sequence analysis.

Finally, we summarize our findings and suggest several directions for future research in Chapter 12. In Appendix D we provide proofs for the theorems and propositions that are included in our analysis.

7.1 Bayesian networks and multi-information

A Bayesian network structure over a set of random variables $\mathbf{X} \equiv \{X_1, \ldots, X_n\}$ is a Directed A-cyclic Graph (DAG) G in which vertices are annotated by names of random variables (see, e.g., [59]). For each variable X_i, we denote by $\mathbf{Pa}^G_{X_i}$ the (potentially empty) set of parents of X_i in G, and by $\mathbf{pa}^G_{X_i}$ a specific assignment to this set of variables. We say that a distribution p is *consistent* with G, if and only if p can be factored in the form:

$$p(x_1, \ldots, x_n) = \prod_i p(x_i \mid \mathbf{pa}^G_{X_i}) \tag{7.1}$$

and use the notation $p \models G$ to denote that.

One of the main concepts that we will deal with is the amount of information that variables $X_1, \ldots, X_n$ contain about each other. As described in Section 1.2.3, a quantity that captures this is the *multi-information* given by Definition 1.2.13. For completeness, we repeat here this definition, in a slightly different form:

$$\begin{aligned} \mathcal{I}(X_1, \ldots, X_n) &= D_{KL}[p(x_1, \ldots, x_n) \| p(x_1) \ldots p(x_n)] \\ &= E_P[\log \frac{p(x_1, \ldots, x_n)}{p(x_1) \ldots p(x_n)}] \, . \end{aligned}$$

Recall that the multi-information captures how close the distribution $p(x_1, \ldots, x_n)$ is to the factored distribution of the marginals. If this quantity is small, we do not lose much by approximating p by the product distribution. Alternatively, it measures the average number of bits that can be gained by a joint compression of the variables versus independent compression. The multi-information is a natural generalization of the pairwise concept of mutual information. Like mutual information, it is non-negative, and equal to zero if and only if all the variables are independent. As shown in [55], it is possible to provide a simple axiomatic derivation for this concept. That is, the multi-information is the *only* function that satisfies the five simple conditions described in Section 1.2.3.

When p has additional known independence relations, we can rewrite the multi-information in terms of the dependencies among the variables:

Proposition 7.1.1 : *Let G be a Bayesian network structure over $\mathbf{X} = \{X_1, \ldots, X_n\}$, and let p be a distribution over $\mathbf{X}$ such that $p \models G$. Then,*

$$\mathcal{I}(\mathbf{X}) = \mathcal{I}[p(\mathbf{x})] = \sum_i I(X_i; \mathbf{Pa}^G_{X_i}) \, . \tag{7.2}$$

That is, the multi-information is the sum of *local* mutual information terms between each variable and its parents. Note that in general, even if $p(\mathbf{x})$ is *not* consistent with G the above sum is well defined. Hence, we state the following definition.

Definition 7.1.2 : The multi-information in $p(\mathbf{x})$ with respect to a given Bayesian network structure G is defined as

$$\mathcal{I}^G \equiv \sum_i I(X_i; \mathbf{Pa}^G_{X_i}) \, , \tag{7.3}$$

where each of the local mutual information terms is calculated using the marginal distributions of $p(\mathbf{x})$.

Note that if p is not consistent with G then in general the real multi-information in p, given by $\mathcal{I}(\mathbf{X})$, is different from $\mathcal{I}^G$. In this case we often want to know how close p is to some distribution which is consistent with G. That is, what the "distance" (or distortion) of p is from its projection onto the sub-space of distributions consistent with G. We define this distortion as

$$D_{KL}[p\|G] \equiv \min_{q \models G} D_{KL}[p\|q] \; .^1 \tag{7.4}$$

The following proposition specifies the form of q for which the minimum is attained.

Proposition 7.1.3: *Let $p(\mathbf{x})$ be a distribution and let G be a DAG . Then*

$$D_{KL}[p\|G] = D_{KL}[p\|q^*] \; , \tag{7.5}$$

where q^* is given by

$$q^*(\mathbf{x}) = \prod_i p(x_i \mid \mathbf{pa}^G_{X_i}) \; . \tag{7.6}$$

Expressed in words, q^* is equivalent to the factorization of p using the conditional independences implied by G. Note that this proposition is a general extension of Proposition 2.1.2. The next proposition provides two possible interpretations of $D_{KL}[p\|G]$, in terms of the structure of G.

Proposition 7.1.4: *Let G be a Bayesian network structure over $\mathbf{X} = \{X_1, \ldots, X_n\}$ where $\mathbf{X} \sim p(\mathbf{x})$. Assume that the order $X_1, \ldots, X_n$ is consistent with the DAG G (i.e., $\mathbf{Pa}^G_{X_i} \subseteq \{X_1, \ldots, X_{i-1}\}$). Then*

$$\begin{aligned} D_{KL}[p\|G] &= \sum_i I(X_i; \{X_1, \ldots, X_{i-1}\} \setminus \{\mathbf{Pa}^G_{X_i}\} \mid \mathbf{Pa}^G_{X_i}) \\ &= \mathcal{I}(\mathbf{X}) - \mathcal{I}^G \; . \end{aligned}$$

Thus, we see that $D_{KL}[p\|G]$ can be expressed as a sum of local conditional information terms, where each term corresponds to a (possible violation of a) Markov independence assumption with respect to the structure of G. If every X_i is independent of $\{X_1, \ldots, X_{i-1}\} \setminus \mathbf{Pa}^G_{X_i}\}$ given $\{\mathbf{Pa}^G_{X_i}\}$ (as implied by G) then $D_{KL}[p\|G]$ becomes zero. As these (conditional) independence assumptions are more extremely violated in p, the corresponding $D_{KL}[p\|G]$ will increase. Recall that the Markov independence assumptions (with respect to a given order) are necessary and sufficient to require the factored form of distributions consistent with G [59]. Therefore, we see that $D_{KL}[p\|G] = 0$ if and only if p is consistent with G.

An alternative interpretation of this measure is given in terms of multi-information terms. Specifically, we see that $D_{KL}[p\|G]$ can be written as the difference between the real multi-information, $\mathcal{I}(\mathbf{X})$, and the multi-information as though $p \models G$, denoted by $\mathcal{I}^G$. Hence, we can think of $D_{KL}[p\|G]$ as the amount of information between the variables that is *not* captured by the dependencies that are implied in the structure of G.

[1] Note that the minimization is over the *second* KL argument, while the first argument remains constant. This is in contrast to the known definition of the *I-projection* [22] of a distribution p on a set of distributions q, given by $q^* = argmin_{q \in Q} D_{KL}[q\|p]$, where here the minimization is over the first KL argument.

Chapter 8

Multivariate Extensions of the IB Method

In this chapter we introduce a general formulation for a multivariate extension of the single-sided IB principle. In the first two sections we develop the multivariate IB principle, and an alternative principle which provides a different interpretation for the method. We discuss the relationships between these two alternatives in Section 8.3, and conclude with several concrete examples in the last section. We will further use these examples in the following chapters.

8.1 Multi-information bottleneck principle

The concept of multi-information allows us to introduce a simple "lift-up" of the original IB variational principle to the multivariate case, using the semantics of Bayesian networks. Given a set of observed (or, input) variables, $\mathbf{X} = \{X_1, \ldots, X_n\}$, instead of one compression variable T, we now specify a set of random variables $\mathbf{T} = \{T_1, \ldots, T_k\}$, which corresponds to different partitions of various subsets of the observed variables. This specification should address two issues. First, loosely speaking, we need to specify "what compresses what". More formally stated, for each subset of $\mathbf{X}$ that we would like to compress, we specify a corresponding subset of the compression variables $\mathbf{T}$. Second, analogous to the original IB problem, we define the solution space in terms of the independences we require between the observed $\mathbf{X}$ variables and the compression $\mathbf{T}$ variables. Recall that for the original IB problem this is achieved through the IB Markovian relation $T \leftrightarrow X \leftrightarrow Y$. As a result, the solution space consists of all the distributions over X, Y, T, such that $p(x, y, t) = p(x, y)p(t|x)$, where the free parameters correspond to the stochastic mapping between X and T. In the multivariate case, the analogous situation would be to define the solution space through a *set* of IB Markovian independence relations, which imply that each compression variable, $T_j \in \mathbf{T}$, is completely defined given the variables it compresses, denoted here as $\mathbf{U}_j \subset \mathbf{X}$.

We achieve these two goals by first introducing a DAG G_{in} over $\mathbf{X} \cup \mathbf{T}$ where the variables in $\mathbf{T}$ are leafs. Given a joint distribution over the observed variables, $p(\mathbf{x})$, G_{in} is defined such that $p(\mathbf{x})$ is consistent with its structure restricted to $\mathbf{X}$. The edges from $\mathbf{X}$ to $\mathbf{T}$ define "what compresses what" and the independences implied by G_{in} correspond to the required set of IB Markovian independence relations. In particular this implies that every T_j is independent of all the other variables, given the variables it compresses, $\mathbf{U}_j = \mathbf{Pa}_{T_j}^{G_{in}} \subset \mathbf{X}$. Hence, the multivariate IB solution space consists of all the distributions over $\mathbf{X} \cup \mathbf{T}$ that satisfy G_{in}. Specifically, the form of these distributions is given by

$$p(\mathbf{x}, \mathbf{t}) = p(\mathbf{x}) \prod_{j=1}^{k} p(t_j \mid \mathbf{pa}_{T_j}^{G_{in}}) \,, \qquad (8.1)$$

where the free parameters correspond to the stochastic mappings $p(t_j \mid \mathbf{pa}_{T_j}^{G_{in}})$. [1] Analogously to the original IB formulation, the information that we would like to minimize is now given by $\mathcal{I}^{G_{in}}$. Since $p(\mathbf{x}, \mathbf{t}) \models G_{in}$ then $\mathcal{I}^{G_{in}} = \mathcal{I}(\mathbf{X}, \mathbf{T})$, i.e., this is the real multi-information in $p(\mathbf{x}, \mathbf{t})$. Minimizing this quantity attempts to make the $\mathbf{T}$ variables as independent of the $\mathbf{X}$ variables as possible. Note that we only modify conditional distributions that refer to variables in $\mathbf{T}$, and we do not modify the dependencies among the original observed $\mathbf{X}$ variables.

Once G_{in} is defined we need to specify the relevant information that we want to preserve. We do that by specifying another DAG , G_{out}. Roughly speaking, G_{out} determines "what predicts what". More formally stated, for each T_j, we define in G_{out} which variables it should predict, or preserve information about. These are simply its children in G_{out}. Thus, using Definition 7.1.2, we may think of $\mathcal{I}^{G_{out}}$ as a measure of how much information the variables in $\mathbf{T}$ maintain about their target variables. This suggests that we should maximize $\mathcal{I}^{G_{out}}$.

The *multivariate IB-functional* can now be written as

$$\mathcal{L}^{(1)}[p(\mathbf{x}, \mathbf{t})] = \mathcal{I}^{G_{in}} - \beta \mathcal{I}^{G_{out}} \ , \tag{8.2}$$

where the variation is done subject to the normalization constraints on the partition distributions, and β is a positive Lagrange multiplier controlling the trade-off. [2] It leads to a tractable formal solution, as we show in the next chapter. Note that this functional is a direct generalization of the original IB-functional, Eq. (2.13). Again, we try to balance between minimizing the compression (multi) information, now defined through $\mathcal{I}^{G_{in}}$, and maximizing the relevant (multi) information, now defined through $\mathcal{I}^{G_{out}}$.

As for the original IB principle the range of β for the multivariate formulation is between 0 to ∞. For $\beta \rightarrow 0$ we concentrate on compression only which yields a trivial solution in which the T_j's are independent of their parents. In other words, in this case each T_j consists of one value to which all the values of $\mathbf{Pa}_{T_j}^{G_{in}}$ are mapped. Hence, all the distinctions between these values (relevant or not) are lost. For $\beta \rightarrow \infty$ we ignore the need for compression and concentrate on maintaining the relevant information terms as high as possible. This, in turn, yields a trivial solution of the opposite extreme, in which each T_j is simply a copy of $\mathbf{Pa}_{T_j}^{G_{in}}$. The interesting cases are in between, where β takes positive final values.

Example 8.1.1 : As a simple example, consider application of the multivariate variational principle with G_{in} and $G_{out}^{(a)}$ of Figure 8.1. G_{in} specifies that T compresses X and $G_{out}^{(a)}$ specifies that we want T to preserve information about Y. For this choice of DAGs, $\mathcal{I}^{G_{in}} = I(T;X) + I(X;Y)$ and $\mathcal{I}^{G_{out}} = I(T;Y)$. The resulting functional is

$$\mathcal{L}^{(1)} = I(X;Y) + I(T;X) - \beta I(T;Y) \ .$$

Since, $I(X;Y)$ is constant, we can ignore it, and we end up with a functional equivalent to that of the original IB principle, given in Eq. (2.13).

8.2 Alternative variational principle

We now describe an alternative and closely related variational principle. This principle is based on approximating distributions with respect to a class defined by the Bayesian network G_{out}, rather than on preservation of multi-information.

[1] For simplicity, we restrict attention to cases where the input distribution $p(\mathbf{x})$ is consistent only with the complete graph over $\mathbf{X}$. Hence, G_{in} restricted to $\mathbf{X}$ must form a complete graph.

[2] Since $\mathcal{I}^{G_{out}}$ typically consists of several mutual information terms (Eq. (7.3)), in principle it is also possible to define a separate Lagrange multiplier for each of these terms. In some situations this option might be useful, for example if for some reason the preservation of one information term is of greater significance than the preservation of the others. Nonetheless, for the sake of simplicity we do not discuss this alternative in the following and leave it for future research.

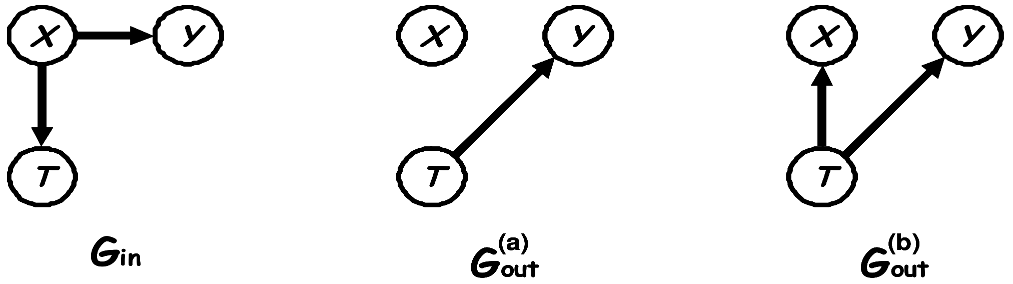

Figure 8.1: The source (left panel) and target networks for the original single-sided IB. The target network for the multivariate IB principle is presented in the middle panel. The target network for the alternative principle is described in the right panel.

We again face the problem of choosing the conditional distributions $p(t_j \mid \mathbf{pa}_{T_j}^{G_{in}})$. Therefore, we must specify our aim in constructing these variables. As with the original IB method, we assume that there are two goals.

On the one hand, we want to compress, or partition, the values of the observed variables. As before the natural multivariate form of this is to minimize the multi-information of $p(\mathbf{x}, \mathbf{t})$, denoted by $\mathcal{I}^{G_{in}}$ (recall that $p \models G_{in}$, therefore $\mathcal{I}^{G_{in}} = \mathcal{I}[p(\mathbf{x}, \mathbf{t})]$).

While in the previous section the second goal was to preserve the multi-information about some (target, relevant) variables, here we think of a *target class* of model distributions, specified by a target Bayesian network. In this interpretation the compressed variables should help us in describing the joint distribution with respect to a different desired structure. We specify this structure by the DAG G_{out}, that now represents which dependencies and independences we would like to impose.

To make this more concrete consider again the two-variable case shown in Figure 8.1. In this example, we are given the distribution of two variables X and Y. The DAG G_{in} specifies that T is a compressed version of X. The ideal situation in our context is when T preserves *all* the information about Y. The following proposition shows that this is equivalent to the situation where T separates between X and Y, [3] which is specified by the DAG $G_{out}^{(b)}$ of Figure 8.1.

Proposition 8.2.1: *Assume that* $T \leftrightarrow X \leftrightarrow Y$*, then* $I(T;Y) = I(X;Y)$ *if and only if* $X \leftrightarrow T \leftrightarrow Y$.

The question now is how to force a construction of $p(t \mid x)$ such that it will lead to the independences that are specified in the target DAG , G_{out}. Note that these G_{in} and G_{out} are, in general, incompatible: Except for trivial cases, we cannot achieve both sets of independences simultaneously. Instead, we aim to come as close as possible to achieving this by a trade-off between the two. We formalize this by requiring that p can be closely *approximated* by a distribution consistent with G_{out}. As previously discussed, a possible information theoretic measure to this approximation is $D_{KL}[p\|G]$, the minimal KL divergence from p to distributions consistent with G_{out}. Recall that $D_{KL}[p\|G_{out}]$ measures the amount of conditional information between variables that are supposed to be conditionally independent in G_{out}. Thus, minimizing $D_{KL}[p\|G_{out}]$ strives to weaken these dependencies as much as possible.

Extending this idea to the general case is straightforward. As before, we introduce a Lagrange multiplier that controls the trade-off between the two objectives. To distinguish it from the previous parameter, we denote this parameter by γ. The functional we want to minimize in this formulation is thus:

$$\mathcal{L}^{(2)}[p(\mathbf{x}, \mathbf{t})] = \mathcal{I}^{G_{in}} + \gamma D_{KL}[p\|G_{out}] \tag{8.3}$$

[3] We say that A *separates* B and C if B and C are conditionally independent given A, i.e., $B \leftrightarrow A \leftrightarrow C$.

where the parameters that we can change during the minimization are again the (normalized) parameters of the conditional distributions $p(t_j \mid \mathbf{pa}_{T_j}^{G_{in}})$.

The range of γ is between 0, in which case we have the trivial (maximally compressed) solution, and ∞, in which we strive to make p as close as possible to G_{out}.

Example 8.2.2: Consider again the example of Figure 8.1 with G_{in} and $G_{out}^{(b)}$. In this case, we have $\mathcal{I}^{G_{in}} = I(X;Y) + I(T;X)$ and $\mathcal{I}^{G_{out}} = I(T;X) + I(T;Y)$. Using Proposition 7.1.4, we have $D_{KL}[p \| G_{out}] = \mathcal{I}^{G_{in}} - \mathcal{I}^{G_{out}}$. Putting these together, we get

$$\mathcal{L}^{(2)} = I(T;X) - \gamma I(T;Y) + (1+\gamma) I(X;Y)$$

As before, by ignoring the constant $I(X;Y)$ term we end up with the original IB-functional (setting $\gamma = \beta$). Thus, we can think of the original IB problem as finding a compression T of X that results in a joint distribution that is as close as possible to the DAG where X and Y are independent given T.

8.3 Relations between the two principles

Going back to the general case, we can apply Proposition 7.1.4 to rewrite the alternative multivariate IB-functional in terms of multi-informations:

$$\mathcal{L}^{(2)} \;=\; \mathcal{I}^{G_{in}} + \gamma(\mathcal{I}^{G_{in}} - \mathcal{I}^{G_{out}}) = (1+\gamma)\mathcal{I}^{G_{in}} - \gamma \mathcal{I}^{G_{out}}$$

which is similar to the functional $\mathcal{L}^{(1)}$ presented in the previous section, under the transformation $\beta = \frac{\gamma}{1+\gamma}$. In this transformation the range $\gamma \in [0, \infty)$ corresponds to the range $\beta \in [0, 1)$. Note that when $\beta = 1$, we have $\mathcal{L}^{(1)} = D_{KL}[p \| G_{out}]$, which is the extreme case of $\mathcal{L}^{(2)}$. Thus, from a mathematical perspective, $\mathcal{L}^{(2)}$ is a special case of $\mathcal{L}^{(1)}$ with the restriction $\beta \leq 1$.

This transformation raises the question of the relation between the two functionals. As we have seen in Example 8.1.1 for each principle we need different versions of G_{out} to reconstruct the single-sided IB-functional. More generally, for a given principle, different choices of G_{out} will yield different optimization problems. Alternatively, given G_{out}, different choices of the variational principle will yield different optimization problems. In the previous example we saw that these two effects can compensate for each other. In other words, using the alternative variational principle with a different choice of G_{out} ends up with the same optimization problem, which in this case is equivalent to the original IB problem.

To further understand the differences between the two principles, we consider the range of solutions for extreme values of β and γ. When $\beta \to 0$ and $\gamma \to 0$, in both formulations we simply minimize $\mathcal{I}^{G_{in}}$. That is, the emphasis is on compression, namely losing information in the transformation from $\mathbf{X}$ to $\mathbf{T}$. In the other extreme case, the two principles differ. When $\beta \to \infty$, minimizing $\mathcal{L}^{(1)}$ is equivalent to maximizing $\mathcal{I}^{G_{out}}$. That is, the emphasis is on preserving information about variables that have parents in G_{out}. For example, in the application of $\mathcal{L}^{(1)}$ in Example 8.1.1 with $G_{out}^{(a)}$, this extreme case results in maximization of $I(T;Y)$. On the other hand, if we apply $\mathcal{L}^{(1)}$ with $G_{out}^{(b)}$, then we maximize $I(T;X) + I(T;Y)$. In this case, when β approaches ∞ information about X will be preserved even if it is irrelevant to Y.

When $\gamma \to \infty$, minimizing $\mathcal{L}^{(2)}$ is equivalent to minimizing $D_{KL}[p \| G_{out}]$. By Proposition 7.1.4 this is equivalent to minimizing the violations of conditional independences implied by G_{out}. Thus, for $G_{out}^{(b)}$, this minimizes $I(X;Y \mid T)$. Using the structure of G_{in} and Proposition 7.1.4, we can write $I(X;Y \mid T) = I(X;Y) - I(T;Y)$, hence this is equivalent to maximizing $I(T;Y)$. If instead we use $G_{out}^{(a)}$, by the same proposition we see that we minimize the information $I(X;Y \mid T) = I(X;Y) + I(T;X) - I(T;Y)$. Thus, we minimize $I(T;X)$ while maximizing $I(T;Y)$. Unlike the application of $\mathcal{L}^{(1)}$ to $G_{out}^{(a)}$, we cannot ignore the term $I(T;X)$.

To summarize, we might loosely say that $\mathcal{L}^{(1)}$ focuses on the edges that are present in G_{out}, while $\mathcal{L}^{(2)}$ focuses on the edges that are *not* present in G_{out} (or more precisely, the conditional independences they imply). This explains the somewhat different intuitions that apply to understanding the solutions found by the variational principles. Thus, although both principles can be applied to any choice of G_{out}, some choices can make more sense for $\mathcal{L}^{(1)}$ than for $\mathcal{L}^{(2)}$, and vice versa.

8.4 Examples: IB variations

We now consider a few examples of these principles applied to different situations. In particular this should help us to further elucidate the relationships between these two formulations.

8.4.1 Parallel IB

We first consider a simple extension of the original IB. Suppose we introduce k compression variables $T_1, \ldots, T_k$ instead of one. As specified in G_{in} of Figure 8.2 (upper panel), all of these variables are stochastic functions of X. In addition, similarly to the original IB, we want $T_1, \ldots, T_k$ to preserve the information X maintains about Y as high as possible. This is specified by the DAG $G_{out}^{(a)}$ in the same figure. We call this example the *parallel* IB, as $T_1, \ldots, T_k$ compress X in "parallel".

Based on these two choices we get, $\mathcal{I}^{G_{in}} = I(X;Y) + \sum_{j=1}^{k} I(T_j;X)$ and $\mathcal{I}^{G_{out}^{(a)}} = I(T_1, \ldots, T_k; Y)$. After dropping the constant term $I(X;Y)$, the Lagrangian $\mathcal{L}^{(1)}$ can be written as

$$\mathcal{L}_a^{(1)} = \sum_{j=1}^{k} I(T_j;X) - \beta I(T_1, \ldots, T_k; Y) \ . \tag{8.4}$$

Thus, we attempt to minimize the information between X and every T_j while maximizing the information all the T_j's preserve together about Y. Using the structure of G_{in} we find that all the T_j's are independent given X. Therefore, we can also rewrite [4]

$$\sum_{j=1}^{k} I(T_j;X) = I(T_1, \ldots, T_k; X) + \mathcal{I}(T_1, \ldots, T_k) \ , \tag{8.5}$$

where $\mathcal{I}(T_1, \ldots, T_k)$ is the multi-information of all the compression variables. Thus, minimizing $\sum_{j=1}^{k} I(T_j;X)$ is equivalent to minimizing $I(T_1, \ldots, T_k; X) + \mathcal{I}(T_1, \ldots, T_k)$. Using this last result we have

$$\mathcal{L}_a^{(1)} = I(T_1, \ldots, T_k; X) + \mathcal{I}(T_1, \ldots, T_k) - \beta I(T_1, \ldots, T_k; Y) \ . \tag{8.6}$$

In other words, another interpretation for the above optimization is that we aim to find $T_1, \ldots, T_k$ that together try to compress X, preserve the information about Y and remain independent of each other as much as possible. In this sense, we can say that we are trying to decompose the information X contains about Y into k "orthogonal" components.

Recall, that using $\mathcal{L}^{(2)}$ we aim at minimizing violation of independences in G_{out}. This suggests that the DAG $G_{out}^{(b)}$ of Figure 8.2 (upper panel) captures our intuitions above. In this DAG , X and Y are independent given every T_j. Moreover, here again $G_{out}^{(b)}$ specifies an additional independence requirement

[4] Proof: $I(T_1, \ldots, T_k; X) = E[\log \frac{p(x,t_1,\ldots,t_k)}{p(x)p(t_1,\ldots,t_k)}] = E[\log \frac{p(x)p(t_1|x)\ldots p(t_k|x)}{p(x)p(t_1,\ldots,t_k)} \cdot \frac{p(t_1)\ldots p(t_k)}{p(t_1)\ldots p(t_k)}] = \sum_{j=1}^{k} I(T_j;X) - \mathcal{I}(T_1, \ldots, T_k)$.

over $T_1, \ldots, T_k$. To see this we examine the functional defined by these specifications. In this case, $\mathcal{I}^{G^{(b)}_{out}} = I(T_1, \ldots, T_k; X) + I(T_1, \ldots, T_k; Y)$. Using Eq. (8.5) the functional $\mathcal{L}^{(2)}$ can be written as

$$\mathcal{L}^{(2)}_b = \sum_{j=1}^{k} I(T_j; X) + \gamma(\mathcal{I}(T_1, \ldots, T_k) - I(T_1, \ldots, T_k; Y)) \ ,$$

which is reminiscent of Eq. (8.6). Again, we attempt to find compressed versions of X that together maximize the information they maintain about Y while remaining independent of each other as possible.

8.4.2 Symmetric IB

We now consider another natural extension of the original IB which we term *symmetric* IB. In this case, we want to compress X into T_X and Y into T_Y such that T_X extracts the information X contains about Y, and at the same time T_Y extracts the information Y contains about X. The DAG G_{in} of Figure 8.2 (middle panel) captures the form of the compression. The choice of G_{out} is less obvious.

One alternative, shown as $G^{(a)}_{out}$ in the figure, attempts to make each of T_X and T_Y sufficient to separate X from Y. As we can see, in this network X is independent of Y (and T_Y) given T_X. Similarly, T_Y separates Y from the other variables. The structure of the network states that T_X and T_Y are dependent of each other. Developing the functional defined by this network, we obtain:

$$\mathcal{L}^{(2)}_a = I(T_X; X) + I(T_Y; Y) - \gamma I(T_X; T_Y) \tag{8.7}$$

Thus, on one hand we attempt to compress, and on the other hand we attempt to make T_X and T_Y as informative about each other as possible. (Note that if T_X is informative about T_Y, then it is also informative about Y.)

Alternatively, we might argue that T_X and T_Y should each compress different aspects of the information between X and Y. This intuition is specified by the target network $G^{(b)}_{out}$ of Figure 8.2 (middle panel). In this network T_X and T_Y are independent of each other, and both are needed to make X and Y conditionally independent. In this sense, our aim is to find T_X and T_Y that capture independent attributes of the connection between X and Y. Indeed, following arithmetic similar to that of the previous example (and using the conditional independences implied by G_{in}), we can write the functional as:

$$\mathcal{L}^{(2)}_b \quad = \quad I(T_X; X) + I(T_Y; Y) - \gamma(I(T_X; Y) + I(T_Y; X) - 2I(T_X; T_Y))$$

That is, we attempt to maximize the information T_X maintains about Y and T_Y about X, and at the same time - in contrast to the previous case - try to *minimize* the information between T_X and T_Y. [5]

8.4.3 Triplet IB

In sequential data, such as natural language text or DNA sequences, an important question is to identify features relevant to predicting a symbol in the sequence. Typically these features are different for “forward prediction” versus “backward prediction”. For example, the textual features that predict the next symbol (word) to be “information” are clearly different from those that predict the *previous* symbol to be “information”. Here we address this issue by extracting features of both types such that their combination is highly informative with respect to predicting a symbol *between* other known symbols.

[5] It is straightforward to extend this example to include k compression variables, instead of two, as we did in the previous parallel IB example. In this case we seek for k compression variables that capture different attributes of the information between X and Y, and try to remain independent of each other as possible.

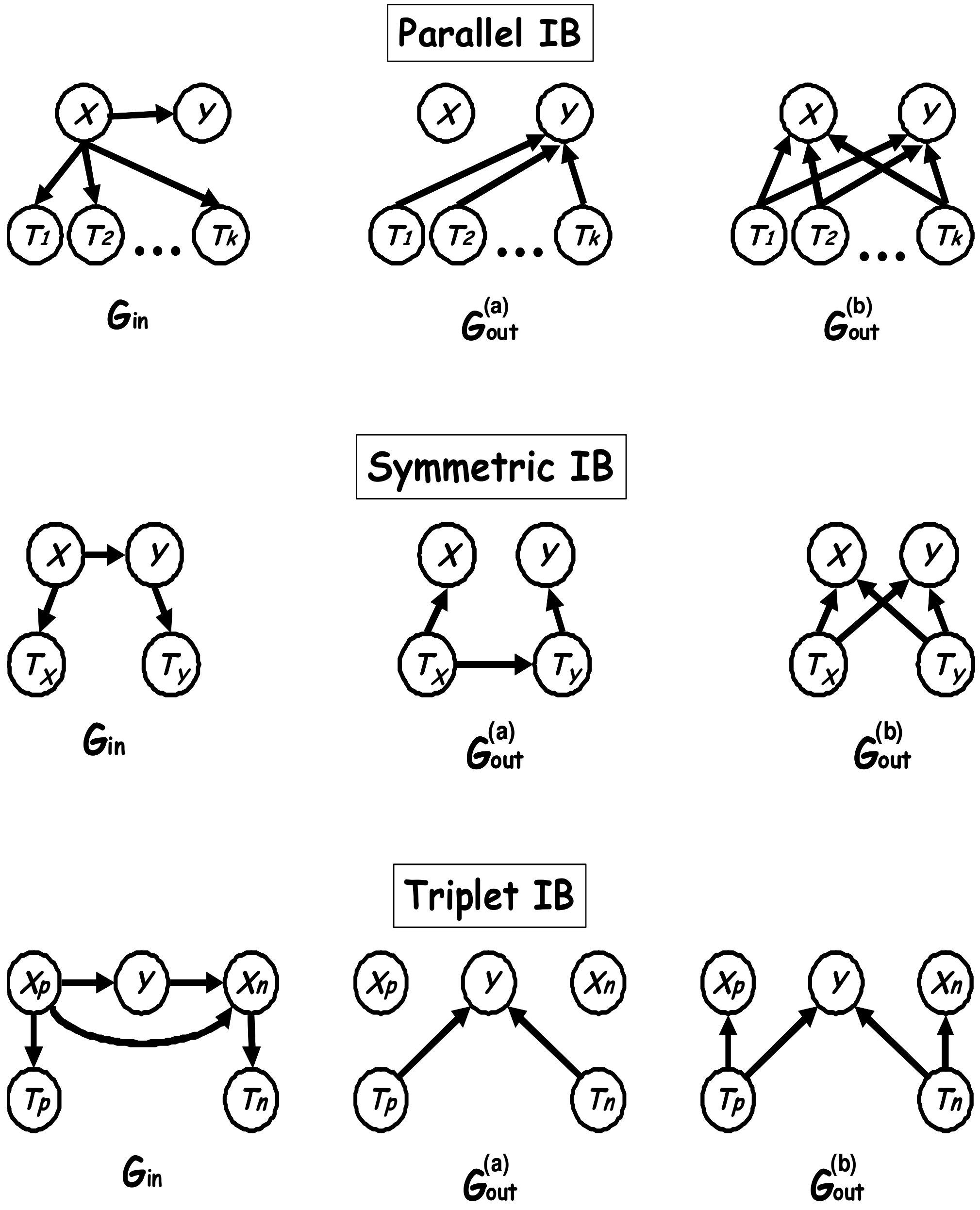

Figure 8.2: Possible source and target DAG s for the symmetric, parallel, and triplet IB examples.

The DAG G_{in} of Figure 8.2 (lower panel) is one way of capturing the form of the compression, where we denote by X_p, Y, X_n the previous, current and next symbol in a given sequence, respectively. In this case, T_p compresses X_p while T_n compresses X_n. For the choice of G_{out} we consider again two alternatives.

First, we simply require that the combination of T_p and T_n will maximally preserve the information X_p and X_n hold about the current symbol Y. This is specified by the DAG $G_{out}^{(a)}$ in the figure. Based on these choices we get,

$$\mathcal{L}_a^{(1)} = I(T_p; X_p) + I(T_n; X_n) - \beta I(T_p, T_n; Y) \ . \tag{8.8}$$

Hence, we are looking for compressed versions of X_p and X_n, that maximally preserve the information about a symbol between them, denoted by Y.

Second, we use the alternative $\mathcal{L}^{(2)}$ principle. Recall that in this case we are interested in satisfying (as much as possible) the conditional independences implied by G_{out}. This suggests that the DAG $G_{out}^{(b)}$ of Figure 8.2 may represent our desired target model. In this network, T_p and T_n are independent, and both are needed to make Y (conditionally) independent of X_p and X_n. Hence, we may see the resulting T_p and T_n partitions as providing compact independent and informative evidences, regarding the value of Y. This specification yields

$$\mathcal{L}_b^{(2)} = I(T_p; X_p) + I(T_n; X_n) - \gamma I(T_p, T_n; Y) \ , \tag{8.9}$$

which is equivalent to Eq. (8.8). In other words, as in Example 8.1.1 we see that by using the alternative variational principle, $\mathcal{L}^{(2)}$, and a different specification of G_{out} we end up with the same optimization problem as by using $\mathcal{L}^{(1)}$. We will term this example the *triplet* IB.

Chapter 9

Characterization of the Solution

In Section 2.4 we saw that it is possible to characterize the form of the optimal solution to the original IB-problem. Can we provide such a characterization to the multivariate IB principles discussed in the previous chapter? It turns out that the answer to this question is positive, as we show below.

Note that a solution to the multivariate IB principle inherently requires a higher level of abstraction. Specifically, it should apply to *any* specification of G_{in} and G_{out}. Hence, we expect to use such a solution as a *recipe*. Once the specification is provided, it induces a concrete solution out of the general form, that characterizes the optimal solution to the optimization problem defined by the choices of G_{in} and G_{out}.

9.1 A formal optimal solution

We assume that G_{in}, G_{out}, and β (or γ) are given. We now want to describe the properties of the distributions $p(t_j \mid \mathbf{pa}_{T_j}^{G_{in}})$ which optimize the trade-off defined by each of the two alternative principles. We present this characterization to the functionals of the form of $\mathcal{L}^{(1)}$. However, we can easily recover the corresponding characterization to functionals of the form $\mathcal{L}^{(2)}$ (using the transformation $\beta = \frac{\gamma}{1+\gamma}$). As we will see later on, the characterization of the optimal solution provides a general extension to the optimal solution of the original IB problem, presented in Section 2.4.

In the presentation of this characterization, we need some additional notational shorthands, given by $\mathbf{U}_j = \mathbf{Pa}_{T_j}^{G_{in}}$, $\mathbf{V}_{T_j} = \mathbf{Pa}_{T_j}^{G_{out}}$, and $\mathbf{V}_{X_i} = \mathbf{Pa}_{X_i}^{G_{out}}$. We also denote $\mathbf{V}_{T_\ell}^{-j} = \mathbf{V}_{T_\ell} \setminus \{T_j\}$ and similarly for $\mathbf{V}_{X_i}^{-j} = \mathbf{V}_{X_i} \setminus \{T_j\}$. To simplify the presentation, we also assume that $\mathbf{U}_j \cap \mathbf{V}_{T_j} = \emptyset$. In addition, we use the notation

$$\begin{aligned} & E_{p(\cdot \mid \mathbf{u}_j)}[D_{KL}[p(y \mid \mathbf{z}, \mathbf{u}_j) \| p(y \mid \mathbf{z}, t_j)]] \\ = & \sum_{\mathbf{z}} p(\mathbf{z} \mid \mathbf{u}_j) D_{KL}[p(y \mid \mathbf{z}, \mathbf{u}_j) \| p(y \mid \mathbf{z}, t_j)] \\ = & E_{p(y, \mathbf{z} \mid \mathbf{u}_j)}[\log \frac{p(y \mid \mathbf{z}, \mathbf{u}_j)}{p(y \mid \mathbf{z}, t_j)}] \end{aligned}$$

where Y is a random variable and $\mathbf{Z}$ is a set of random variables. Note that this term implies averaging over all values of Y and $\mathbf{Z}$ using the conditional distribution. In particular, if Y or $\mathbf{Z}$ intersects with $\mathbf{U}_j$, then only the values consistent with $\mathbf{u}_j$ have positive weights in this averaging. Also note that if $\mathbf{Z}$ is empty, then this term reduces to the standard KL divergence between $p(y \mid \mathbf{u}_j)$ and $p(y \mid t_j)$.

The main result of this chapter is as follows.

Theorem 9.1.1 : *Assume that $p(\mathbf{x})$, G_{in}, G_{out}, and β are given and that $p(\mathbf{x}, \mathbf{t}) \models G_{in}$. The conditional* *distributions* $\{p(t_j \mid \mathbf{u}_j)\}_{j=1}^{k}$ *are a stationary point of* $\mathcal{L}^{(1)}[p(\mathbf{x},\mathbf{t})] = \mathcal{I}^{G_{in}} - \beta\mathcal{I}^{G_{out}}$ *if and only if*

$$p(t_j \mid \mathbf{u}_j) = \frac{p(t_j)}{Z_{T_j}(\mathbf{u}_j,\beta)} e^{-\beta d(t_j,\mathbf{u}_j)},\ \forall\, t_j \in \mathcal{T}_j,\ \forall\, \mathbf{u}_j \in \boldsymbol{\mathcal{U}}_j\ , \tag{9.1}$$

where $Z_{T_j}(\mathbf{u}_j,\beta)$ *is a normalization function, and*

$$\begin{aligned} d(t_j,\mathbf{u}_j) \equiv & \sum_{i:T_j \in \mathbf{V}_{X_i}} E_{p(\cdot|\mathbf{u}_j)}[D_{KL}[p(x_i \mid \mathbf{v}_{X_i}^{-j},\mathbf{u}_j)\|p(x_i \mid \mathbf{v}_{X_i}^{-j},t_j)]] \\ & + \sum_{\ell:T_j \in \mathbf{V}_{T_\ell}} E_{p(\cdot|\mathbf{u}_j)}[D_{KL}[p(t_\ell \mid \mathbf{v}_{T_\ell}^{-j},\mathbf{u}_j)\|p(t_\ell \mid \mathbf{v}_{T_\ell}^{-j},t_j)]] \\ & \qquad\qquad + D_{KL}[p(\mathbf{v}_{T_j} \mid \mathbf{u}_j)\|p(\mathbf{v}_{T_j} \mid t_j)]\ . \end{aligned} \tag{9.2}$$

Note that the first summation is over all the variables X_i such that T_j is aimed at preserving information about. The second summation is over all the variables T_ℓ such that T_j is suppose to preserve information about. The last term corresponds to a situation where information should also be maintained about T_j (by $\mathbf{V}_{T_j}$).

The essence of this theorem is that it defines $p(t_j \mid \mathbf{u}_j)$ in terms of the *multivariate relevant-distortion* $d(t_j,\mathbf{u}_j)$. This distortion measures the degree of proximity of the conditional distributions in which $\mathbf{u}_j$ is involved into these where we replace $\mathbf{u}_j$ with t_j. In other words, we can understand this as measuring how well t_j performs as a "representative" of the particular assignment $\mathbf{u}_j$. As this representative behaves more similarly to $\mathbf{u}_j$, $d(t_j,\mathbf{u}_j)$ becomes smaller, which in turn increases the membership probability, $p(t_j \mid \mathbf{u}_j)$.

As in the single-sided IB problem, the above theorem also allows us to understand the role of β. When β is small, each conditional distribution, $p(t_j \mid \mathbf{u}_j)$ is diffused, since β reduces the differences between the distortions for different values of T_j. On the other hand, when β is large, the exponential term acts as a "soft-max" gate, and most of the conditional probability mass will be assigned to the value t_j with the smallest distortion. Moreover, in the limit $\beta \to \infty$ this value will contain *all* the probability mass, i.e., the above stochastic mapping will become deterministic. This behavior also matches our understanding that when β is small, most of the emphasis is on compressing the input variables $\mathbf{U}_j$ into T_j. When β is large, most of the emphasis is on predicting the target variables of T_j, as specified by G_{out}.

Lastly, note that as in the original IB problem, the effective (multivariate) distortion measure, $d(t_j,\mathbf{u}_j)$, emerges directly from the variational principle $\mathcal{L}^{(1)}$, rather then being assumed in advance in any way. In other words, this is the correct distortion measure to this multivariate principle.

9.2 Examples

First, as a simple sanity-check, we reconsider Example 8.1.1, where we formulate the single-sided IB problem using the specification of G_{in} and $G_{out}^{(a)}$ of Figure 8.1, and the functional $\mathcal{L}^{(1)}$. For these choices it is easy to verify that the multivariate relevant-distortion (Eq. (9.2)) simply amounts to

$$d(t,x) = D_{KL}[p(y \mid x)\|p(y \mid t)]\ , \tag{9.3}$$

which is in full analogous to Eq. (2.16), as required. We further use the general form of Eq. (9.2) to obtain the effective distortion in our additional IB-like variations.

Example 9.2.1: We start by reconsidering the parallel IB case of $G_{out}^{(a)}$ in Figure 8.2, Section 8.4.1. Applying the theorem to the corresponding $\mathcal{L}^{(1)}$ (Eq. (8.4)), we see that the distortion term for every T_j is

$$d(t_j,x) = E_{p(\cdot|x)}[D_{KL}[p(y \mid \mathbf{t}^{-j},x\|p(y \mid \mathbf{t}^{-j},t_j)]]\ , \tag{9.4}$$

where we used the notation $\mathbf{T}^{-j} = \mathbf{T} \setminus \{T_j\}$. This distortion term corresponds to the information of Y and $\mathbf{T} = \{T_1, \ldots, T_k\}$. We see that $p(t_j \mid x)$ increases when the predictions of Y given t_j are similar to those given x (when averaging over $\mathbf{t}^{-j}$).

Example 9.2.2: Consider now the symmetric IB case of $G_{out}^{(a)}$ in Figure 8.2, Section 8.4.2. By dropping the edge between T_X and X and the edge between T_Y and Y we get a different specification of G_{out}. Using the first variational principle for this specification, we get $\mathcal{L}^{(1)} = I(T_X; X) + I(T_Y; Y) - \beta I(T_X; T_Y)$, which is equivalent to Eq. (8.7). Applying the theorem for this functional (and G_{out}) we have

$$d(t_X, x) = E_{p(\cdot|x)}[D_{KL}[p(t_Y \mid x) \| p(t_Y \mid t_X)]] \ . \tag{9.5}$$

Thus, T_X attempts to make predictions as similar to those of X about T_Y (and similarly T_Y attempts to make predictions as similar to those of Y about T_X).

Example 9.2.3 : Last, we consider the triplet IB of $G_{out}^{(a)}$ in the lower panel of Figure 8.2, Section 8.4.3. Applying the theorem again, we have the distortion term for T_p:

$$d(t_p, x_p) = E_{p(\cdot|x_p)}[D_{KL}[p(y \mid t_n, x_p) \| p(y \mid t_n, t_p)]] \ . \tag{9.6}$$

This term corresponds to the information of Y and T_p, T_n. We see that $p(t_p \mid x_p)$ increases when the predictions about Y given by t_p are similar to those given by x_p (when averaging over T_n). The distortion term for T_n is defined analogously.

Chapter 10

Multivariate IB Algorithms

Similarly to the single-sided IB-functional, the multivariate IB-functional is not convex with all of its arguments simultaneously. Hence, different heuristics must be employed to obtain at least locally optimal solutions. In this chapter we show that all the four algorithmic approaches suggested to the original IB problem in Chapter 3 are naturally extended into the multivariate case. In particular this allows to construct solutions in practice to different multivariate IB problems.

The organization of this chapter is similar to that of Chapter 3. That is, in the first section we present the extension to the iIB algorithm. Next, we describe the extensions to the dIB and the aIB algorithm, and finally in Section 10.4 we present the multivariate sIB algorithm. For completeness, some of the descriptions used in Chapter 3 are used again in the following. However, as regard to combining different algorithms and the relations between these algorithms, the discussion in Section 3.5 and Section 3.6 is immediately extended into the multivariate scenario, hence we do not repeat it here.

In the following presentation we concentrate on the variational principle $\mathcal{L}^{(1)}$. Again, applying the same algorithms for $\mathcal{L}^{(2)}$ is straightforward.

10.1 Iterative optimization algorithm: multivariate iIB

We start with the case where β is fixed. In this case, following the strategy suggested in the original iIB algorithm we simply apply the fixed point equations given in Eq. (9.1). Thus, we use an iterative algorithm, that at the m'th iteration maintains the conditional distributions $\{p^{(m)}(t_j \mid \mathbf{u}_j)\}_{j=1}^{k}$. At the $m+1$'th iteration, the algorithm applies an update step:

$$p^{(m+1)}(t_j \mid \mathbf{u}_j) \leftarrow \frac{p^{(m)}(t_j)}{Z_{T_j}^{(m+1)}(\mathbf{u}_j, \beta)} e^{-\beta d^{(m)}(t_j, \mathbf{u}_j)} \tag{10.1}$$

where $p^{(m)}(t_j)$ and $d^{(m)}(t_j, \mathbf{u}_j)$ are computed from $p(\mathbf{x}, \mathbf{t})$ with respect to the conditional probabilities $\{p^{(m)}(t_j \mid \mathbf{u}_j)\}_{j=1}^{k}$, and using the conditional independences implied by G_{in}.

There are two main variants of this algorithm. In the *synchronous* variant, we apply the update step for all the conditional distributions in each iteration. That is, each conditional probability $p(t_j \mid \mathbf{u}_j)$ is updated by computing the distortions based on the conditional probabilities of the previous iteration. In the *asynchronous* variant, we choose one variable T_j, and perform the update only for this variable. For all $\ell \neq j$, we set $p^{(m+1)}(t_\ell \mid \mathbf{u}_\ell) = p^{(m)}(t_\ell \mid \mathbf{u}_\ell)$. The main difference between the two variants is that the update of T_j in the asynchronous case incorporates the implications of the updates of all its "preceding" variables. Additionally, it seems that in the general case, only the *asynchronous* variant is guaranteed to converge to a (locally) optimal solution. This is specified by the following theorem.

```
Input:
    Joint distribution $p(\mathbf{x})$ .
    Trade-off parameter $\beta$ .
    Source DAG : $G_{in}$, with leafs $T_j,\ j = 1:k$ , and Target DAG : $G_{out}$ .
    Cardinality parameters $M_j,\ j = 1:k,$ and convergence parameter $\varepsilon$ .

Output:
    A (typically "soft") partition $T_j$ of $\mathcal{U}_j$ into $M_j$ clusters $\forall j = 1:k$ .

Initialization:
    Randomly initialize $p(t_j \mid \mathbf{u}_j)\ \forall j = 1:k$ .

While True

    For $j = 1:k$ ,

        • $p^{(m+1)}(t_j \mid \mathbf{u}_j) \leftarrow \frac{p^{(m)}(t_j)}{Z^{(m+1)}_{T_j}(\mathbf{u}_j,\beta)} e^{-\beta d^{(m)}(t_j,\mathbf{u}_j)}$ , $\forall\, t_j \in \mathcal{T}_j,\ \forall\, \mathbf{u}_j \in \mathcal{U}_j$ .

        • $p^{(m+1)}(t_j) \leftarrow \sum_{\mathbf{u}_j} p^{(m+1)}(t_j \mid \mathbf{u}_j) p(\mathbf{u}_j)$ , $\forall\, t_j \in \mathcal{T}_j$ .

        • Update all the distributions in $d(t_j, \mathbf{u}_j)$ that explicitly involve $T_j$ ,
          using the independences implied by $G_{in}$ and $p^{(m+1)}(t_j \mid \mathbf{u}_j)$ .

    If $\forall\, j = 1:k,\ \forall\, \mathbf{u}_j \in \mathcal{U}_j,\ \ JS_{\frac{1}{2},\frac{1}{2}}[p^{(m+1)}(t_j \mid \mathbf{u}_j), p^{(m)}(t_j \mid \mathbf{u}_j)] \le \varepsilon$ ,
        Break.
```

Figure 10.1: Pseudo-code of the multivariate iterative IB algorithm (multivariate iIB), with *asynchronous* updates. JS denotes the Jensen-Shannon divergence (Definition 1.2.17). In principle we repeat this procedure for different initializations, and choose the solution which minimizes $\mathcal{L} = \mathcal{I}^{G_{in}} - \beta \mathcal{I}^{G_{out}}$.

Theorem 10.1.1 : *Asynchronous iterations of the fixed-point equations given in Eq. (10.1) converge to a stationary point of the multivariate IB-functional, $\mathcal{L}^{(1)}$.*

Note that this theorem extends the convergence theorem of the single-sided iIB algorithm (Theorem 3.1.1). Moreover, the proof technique is based on the proof of that theorem, hence we delay it to Appendix D. In Figure 10.1 we present Pseudo-code for this asynchronous variant which we will term *multivariate* iIB.

The question of how to initialize this procedure, which is evident for the original iIB algorithm, might be even more acute in the multivariate case. Again, different initializations in general lead to different locally optimal solutions. Moreover, exploring a hierarchy of solutions for different β values is clearly desirable in some cases. To address these issues we present in the next section a multivariate deterministic annealing-like procedure, extending the original dIB algorithm.

10.2 Deterministic annealing-like algorithm: multivariate dIB

Recall that a deterministic annealing procedure works by iteratively increasing the parameter β and then adapting the solution for the previous value of β to the new one. This allows the algorithm to "track" the changes in the solution as the system shifts its preferences from compression to prediction. When $\beta \to 0$, the optimization problem tends to make each T_j independent of its parents. At this point the solution consists of essentially one cluster for each T_j which is not predictive about any other variable. As we increase β, at some (critical) point the values of some T_j diverge and show two different behaviors. Successive increases of β will reach additional phase transitions in which additional splits of some values of the T_j's emerge. Our goal is to identify these cluster bifurcations and eventually record for each T_j a bifurcating tree that traces the sequence of solutions at different values of β (see, for example, Figure 11.1).

To detect these bifurcations we adopt the method of the single-sided dIB algorithm to multiple variables. At each step, we take the solution from the previous β value we considered and construct an initial problem in which we *duplicate* each value of every T_j. Thus, we need to specify the conditional membership probabilities of these duplicated values. Suppose that t_j^ℓ and t_j^r are two such duplications of some value $t_j \in \mathcal{T}_j$. Then we set $p^*(t_j^\ell \mid \mathbf{u}_j) = p(t_j \mid \mathbf{u}_j)\left(\frac{1}{2} + \alpha\hat{\epsilon}(t_j, \mathbf{u}_j)\right)$ and $p^*(t_j^r \mid \mathbf{u}_j) = p(t_j \mid \mathbf{u}_j)\left(\frac{1}{2} - \alpha\hat{\epsilon}(t_j, \mathbf{u}_j)\right)$, where $\hat{\epsilon}(t_j, \mathbf{u}_j)$ is a (stochastic) noise term randomly drawn out of $U[-\frac{1}{2}, \frac{1}{2}]$ and $\alpha > 0$ is a (small) scale parameter. Thus, each copy t_j^ℓ and t_j^r is a slightly perturbed version of t_j. If β is high enough, this random perturbation suffices to allow the two copies of t_j to diverge. If β is too small to support such bifurcation, both perturbed versions will collapse to the same solution.

Given this initial point, we simply apply the (asynchronous) multivariate iIB algorithm. After convergence is attained, if the behavior of t_j^ℓ and t_j^r is "sufficiently different" [1] then we declare that the value t_j has split, and incorporate t_j^ℓ and t_j^r into the hierarchy we construct for T_j. Finally, we increase β and repeat the whole process. We will term this algorithm *multivariate* dIB. A Pseudo-code is given in Figure 10.2.

As in its original single-sided variant, there are some technical difficulties with applying this algorithm. For example, the parameters $d_{min}^{(j)}, \ j = 1:k,$ that control the detection of cluster splits, need to be tuned. As before, it is not clear whether these parameters should be fixed during the process, where a possible alternative is to set them as a function of β (see, e.g., Section 11.2.1). Additionally, one needs to tune the rate of increasing β otherwise cluster splits might be "skipped". Last, as for the original dIB algorithm, the duplication process is stochastic in nature (and involves additional parameters) which in principle is not desirable. In the following section we describe a simpler (although approximated) approach, which extends the single-sided aIB algorithm described in Section 3.3.

10.3 Agglomerative algorithm: multivariate aIB

Following the preliminary work in [73], we now present in detail an extension of the aIB algorithm into the context of multivariate IB problems. For consistency with [73] and with Section 3.3 we examine the problem of *maximizing*

$$\mathcal{L}_{max} = \mathcal{I}^{G_{out}} - \beta^{-1} \cdot \mathcal{I}^{G_{in}} , \tag{10.2}$$

which is clearly equivalent to minimizing the functional $\mathcal{L}^{(1)}$ defined by Eq. (8.2).

We consider procedures that start with a set of clusters (i.e., values) in each T_j (usually the most fine-grained solution we can consider where $T_j = \mathbf{U}_j$) and then iteratively reduce the cardinality of one of the T_j's by *merging* two values, t_j^ℓ and t_j^r of T_j into a single value $\bar{t}_j$. To formalize this notion we need to define the membership probability of a new cluster $\bar{t}_j$, resulting from merging $\{t_j^\ell, t_j^r\} \Rightarrow \bar{t}_j$ in T_j. Similarly to Eq. (3.8), this is done through:

$$p(\bar{t}_j \mid \mathbf{u}_j) = p(t_j^\ell \mid \mathbf{u}_j) + p(t_j^r \mid \mathbf{u}_j) \ . \tag{10.3}$$

Thus, we view the event $\bar{t}_j$ as the union of the events t_j^ℓ and t_j^r.

For the following analysis we need the next definition.

[1] Specifically, we denote by $\mathbf{Ne}_{T_j}^{G_{out}}$ the set of T_j's neighbors in G_{out}, and consider the divergence between $p(\mathbf{ne}_{T_j}^{G_{out}} \mid t_j^\ell)$ and $p(\mathbf{ne}_{T_j}^{G_{out}} \mid t_j^r)$. Other techniques are also plausible.

```
Input:
    Similar to the multivariate iIB.
    Additional Parameters: $\alpha,\ \varepsilon_\beta$, and $d_{min}^{(j)},\ j = 1 : k$ .

Output:
    (Typically “soft”) partitions $T_j$ of $\boldsymbol{\mathcal{U}}_j$ into $m = 1, \ldots, M_j$ clusters $\forall j = 1 : k$ .

Initialization:
    $\beta \leftarrow 0$
    For $j = 1 : k$
        $\mathcal{T}_j \leftarrow \{t_j\},\ p(t_j \mid \mathbf{u}_j) = 1$ .

Main Annealing Loop:
    $\beta \leftarrow f(\beta, \varepsilon_\beta)$

    Duplicate clusters:
    For $j = 1 : k,\ \forall\ t_j \in \mathcal{T}_j$ and $\forall\ \mathbf{u}_j \in \boldsymbol{\mathcal{U}}_j$,
        Randomly draw $\hat{\epsilon}(t_j, \mathbf{u}_j) \sim U[-\frac{1}{2}, \frac{1}{2}]$ and define:
        $p^*(t_j^\ell \mid \mathbf{u}_j) = p(t_j \mid \mathbf{u}_j)\left(\frac{1}{2} + \alpha\hat{\epsilon}(t_j, \mathbf{u}_j)\right)$
        $p^*(t_j^r \mid \mathbf{u}_j) = p(t_j \mid \mathbf{u}_j)\left(\frac{1}{2} - \alpha\hat{\epsilon}(t_j, \mathbf{u}_j)\right)$

    Apply multivariate iIB using the duplicated clusters set as initialization.

    Check for Splits:
    For $j = 1 : k\ , \forall\ t_j \in \mathcal{T}_j$ ,
        If $JS_{\frac{1}{2},\frac{1}{2}}[p(\mathbf{ne}_{T_j}^{G_{out}} \mid t_j^\ell), p(\mathbf{ne}_{T_j}^{G_{out}} \mid t_j^r)] \geq d_{min}^{(j)}$ ,
            $\mathcal{T}_j \leftarrow \{\mathcal{T}_j \setminus \{t_j\}\} \cup \{t_j^\ell, t_j^r\}$

    If $\forall\ j = 1 : k,\ |\mathcal{T}_j| \geq M_j$, return.
```

Figure 10.2: Pseudo-code of the multivariate deterministic annealing-like algorithm (multivariate dIB). JS denotes the Jensen-Shannon divergence (Definition 1.2.17). $\mathbf{Ne}_{T_j}^{G_{out}}$ denotes the neighbors of T_j in G_{out} (i.e., parents or direct descendants). $f(\beta, \varepsilon_\beta)$ is a simple function used to increment β based on its current value and on some scaling parameter ε_β.

Definition 10.3.1: The *conditional merger distribution* of the merger $\{t_j^\ell, t_j^r\} \Rightarrow \bar{t}_j$ in T_j is defined by

$$\Pi_{\mathbf{z}} = \{\pi_{\ell,\mathbf{z}},\ \pi_{r,\mathbf{z}}\} = \{\frac{p(t_j^\ell \mid \mathbf{z})}{p(\bar{t}_j \mid \mathbf{z})}, \frac{p(t_j^r \mid \mathbf{z})}{p(\bar{t}_j \mid \mathbf{z})}\} . \tag{10.4}$$

Note that if $\mathbf{Z} = \emptyset$ then $\Pi_{\mathbf{z}} = \Pi = \{\frac{p(t_j^\ell)}{p(\bar{t}_j)}, \frac{p(t_j^r)}{p(\bar{t}_j)}\}$, as in Eq. (3.11).

Given the membership probabilities, at each step we can draw the connection between T_j and the other variables. This is done using the following proposition which is based on the conditional independence assumptions implied by G_{in}, and extends Proposition 3.3.1.

Proposition 10.3.2: *Let* $\mathbf{Y}, \mathbf{Z} \subset \mathbf{X} \cup \mathbf{T}$. *Then,*

$$p(\mathbf{z}, \bar{t}_j) = p(\mathbf{z}, t_j^\ell) + p(\mathbf{z}, t_j^r) \ , \tag{10.5}$$

and

$$p(\mathbf{y} \mid \mathbf{z}, \bar{t}_j) = \pi_{\ell,\mathbf{z}} \cdot p(\mathbf{y} \mid \mathbf{z}, t_j^\ell) + \pi_{r,\mathbf{z}} \cdot p(\mathbf{y} \mid \mathbf{z}, t_j^r) \ . \tag{10.6}$$

In particular, this proposition allows us to evaluate all the predictions defined in G_{out} and all the information terms in $\mathcal{L}_{max}$ that involve T_j. Additionally, an immediate corollary of this proposition is that $\Pi_{\mathbf{z}}$ is indeed a proper normalized distribution.

To apply an agglomerative procedure we need to characterize each merger "cost". As before, this cost is given by the reduction in $\mathcal{L}_{max}$ do to some merger. Let T_j^{bef} and T_j^{aft} denote the random variables that correspond to T_j, before and after a merger in T_j, respectively. Thus, the corresponding values of $\mathcal{L}_{max}$ are calculated based on T_j^{bef} and T_j^{aft}, and the merger cost is then given by

$$\Delta\mathcal{L}_{max}(t_j^\ell, t_j^r) = \mathcal{L}_{max}^{bef} - \mathcal{L}_{max}^{aft} \ . \tag{10.7}$$

The greedy procedure evaluates all the potential mergers (for each T_j) and then applies the one that minimizes $\Delta\mathcal{L}_{max}(t_j^\ell, t_j^r)$. This is repeated until all the variables in $\mathbf{T}$ degenerate into trivial clusters. The resulting set of trees describe a range of solutions at all the different resolutions.

10.3.1 Multivariate local merging criteria

The above procedure requires at every step to calculate $\approx O(|\mathcal{T}_j|^2)$ merger-costs for every T_j. A direct calculation of all these costs, using Eq. (10.7) is typically unfeasible. However, as in the original aIB algorithm, it turns out that one may calculate $\Delta\mathcal{L}_{max}(t_j^\ell, t_j^r)$ while examining only the distributions that involve t_j^ℓ and t_j^r directly. This is specified in the following theorem which generalizes the corresponding result of Section 3.3.1.

Theorem 10.3.3: *Let* $t_j^\ell, t_j^r \in \mathcal{T}_j$ *be two clusters. Then,*

$$\Delta\mathcal{L}_{max}(t_j^\ell, t_j^r) = p(\bar{t}_j) \cdot \bar{d}(t_j^\ell, t_j^r) \ , \tag{10.8}$$

where

$$\begin{aligned}\bar{d}(t_j^\ell, t_j^r) \equiv & \sum_{i: T_j \in \mathbf{V}_{X_i}} E_{p(\cdot \mid \bar{t}_j)}[JS_{\Pi_{\mathbf{v}_{X_i}^{-j}}}[p(x_i \mid t_j^\ell, \mathbf{v}_{X_i}^{-j}), p(x_i \mid t_j^r, \mathbf{v}_{X_i}^{-j})]] \\ & + \sum_{\ell: T_j \in \mathbf{V}_{T_\ell}} E_{p(\cdot \mid \bar{t}_j)}[JS_{\Pi_{\mathbf{v}_{T_\ell}^{-j}}}[p(t_\ell \mid t_j^\ell, \mathbf{v}_{T_\ell}^{-j}), p(t_\ell \mid t_j^r, \mathbf{v}_{T_\ell}^{-j})]] \\ & + JS_\Pi[p(\mathbf{v}_{T_j} \mid t_j^\ell), p(\mathbf{v}_{T_j} \mid t_j^r)] - \beta^{-1} \cdot JS_\Pi[p(\mathbf{u}_j \mid t_j^\ell), p(\mathbf{u}_j \mid t_j^r)] \ .\end{aligned} \tag{10.9}$$

There is a natural analogy between this merging criterion and the effective distortion measure that controls the multivariate iIB algorithm (Eq. (9.2)). As in the single-sided case, while for the multivariate iIB the optimization is governed by the KL divergences between data and cluster centroids, for the multivariate aIB algorithm the optimization is controlled through the JS divergences. Specifically, the merger cost is (again) a multiplication of the "weight" of the merger components, $p(\bar{t}_j)$, with their "distance" given by $\bar{d}(t_j^\ell, t_j^r)$. Note that due to the properties of the JS divergence this "distance" is symmetric but it is not a metric. In addition, the last term has the opposite sign to the first three terms. Thus, the "distance" between two clusters is a trade-off between these two factors. Roughly speaking, we may say that it is minimized for pairs which give similar predictions about the variables connected with T_j in G_{out} (the variables that T_j should predict), and have different predictions, or minimum overlap about the variables connected with T_j in G_{in} (the variables that T_j should compress).

Next, we note that after applying a merger, only a small portion of the other merger costs change. The following proposition characterizes these costs.

Proposition 10.3.4: *The merger $\{t_j^\ell, t_j^r\} \Rightarrow \bar{t}_j$ in T_j can change the cost $\Delta\mathcal{L}_{max}(t_s^\ell, t_s^r)$ only if $p(\bar{t}_j, \bar{t}_s) > 0$ and T_j, T_s co-appear in some information term in $\mathcal{I}^{G_{out}}$.*

This proposition is especially useful when we consider "hard" clustering where T_j is a deterministic function of $\mathbf{U}_j$. In this case, $p(\bar{t}_j, \bar{t}_s)$ is often zero (especially when T_j and T_s compress similar variables, i.e., $\mathbf{U}_j \cap \mathbf{U}_s \neq \emptyset$). In particular, after the merger $\{t_j^\ell, t_j^r\} \Rightarrow \bar{t}_j$, we do not have to reevaluate merger costs of other values of T_j, except for mergers of $\bar{t}_j$ with each of these values.

In the case of "hard" clustering we also have $I(T_j; \mathbf{U}_j) = H(T_j)$. Therefore, as in the single-sided case, increasing β^{-1} results in a tendency to look for less balanced "hard" partitions. Additionally (as already mentioned in Section 3.3.1), in this case the last term in $\bar{d}(t_j^\ell, t_j^r)$ is simplified through $JS_\Pi[p(\mathbf{u}_j \mid t_j^\ell), p(\mathbf{u}_j \mid t_j^r)] = H[\Pi]$. For brevity, in the rest of this section we focus on this simpler case of "hard" clustering. We emphasize, though, that all of the above analysis holds for "soft" clustering as well, hence in principle it is possible to apply this agglomerative procedure over "soft" partitions. Moreover, as shown in Section 3.6, the obtained "hard" partitions can be used as a platform to find "soft" clustering solutions through a process of "reverse annealing".

10.3.2 Examples

We now briefly consider our previous examples using the general result of Theorem 10.3.3. We first consider the original IB problem, specified in our formulation through G_{in} and $G_{out}^{(a)}$ in Figure 8.1. The merger cost in this case is given by,

$$\Delta\mathcal{L}_{max}(t^l, t^r) = p(\bar{t}) \cdot (JS_\Pi[p(y \mid t^l), p(y \mid t^r)] - \beta^{-1} H[\Pi]) \ , \tag{10.10}$$

which is analogous to Eq. (3.14), as required.

Considering the parallel IB described by the two left upper panels of Figure 8.2 we find that the merger cost for every T_j is given by,

$$\Delta\mathcal{L}_{max}(t_j^\ell, t_j^r) = p(\bar{t}_j) \cdot (E_{p(\cdot|\bar{t}_j)}[JS_{\Pi_{\mathbf{t}^{-j}}}[p(y \mid \mathbf{t}^{-j}, t_j^\ell), p(y \mid \mathbf{t}^{-j}, t_j^r)]] - \beta^{-1} H[\Pi]) \ , \tag{10.11}$$

where again we used $\mathbf{T}^{-j} = \mathbf{T} \setminus \{T_j\}$.

Finally, we consider the symmetric IB described in the two left middle panels of Figure 8.2, and the alternative variational principle (Eq. (8.3)). As already mentioned (Example 9.2.2), equivalently we may drop the edges between T_X and X and between T_Y and Y (in G_{out}), and use the first variational principle (Eq. (8.2)). Having done that we obtain

$$\Delta\mathcal{L}_{max}(t_X^\ell, t_X^r) = p(\bar{t}_X) \cdot (JS_\Pi[p(t_Y \mid t_X^\ell), p(t_Y \mid t_X^r)] - \beta^{-1} H[\Pi]) \ , \tag{10.12}$$

```
Input:
    Joint distribution $p(\mathbf{x})$ .
    Trade-off parameter $\beta$ .
    Source DAG : $G_{in}$, with leafs $T_j,\ j = 1 : k$ , and Target DAG : $G_{out}$ .

Output:
    Partitions $T_j$ of $\mathcal{U}_j$ into $m = 1, \ldots, |\mathcal{U}_j|$ clusters $\forall j = 1 : k$ .

Initialization:
    $T_j \leftarrow \mathbf{U}_j\ , \forall j = 1 : k$ .
    For $j = 1 : k$
        $\forall\ t_j^\ell, t_j^r \in \mathcal{T}_j$ calculate $\Delta\mathcal{L}_{max}(t_j^\ell, t_j^r) = p(\bar{t}_j) \cdot \bar{d}(t_j^\ell, t_j^r)$

Main Loop:
    While $\exists\ j,\ |\mathcal{T}_j| > 1$
        $\{j, \ell, r\} = argmin_{j', \ell', r'} \Delta\mathcal{L}_{max}(t_{j'}^{\ell'}, t_{j'}^{r'})$
        Merge $\{t_j^\ell, t_j^r\} \Rightarrow \bar{t}_j$ in $T_j$
        Update $\Delta\mathcal{L}_{max}(t_{j'}^{\ell'}, t_{j'}^{r'})$ costs w.r.t. $\bar{t}_j$ (only for costs that need an update)
```

Figure 10.3: Pseudo-code of the multivariate agglomerative IB algorithm (multivariate aIB).

and an analogous expression for mergers in T_Y.

Applying the same theorem for the last example of the triplet IB is left to the interested reader. A Pseudo-code of the general procedure is given in Figure 10.3.

10.4 Sequential optimization algorithm: multivariate sIB

The main disadvantages of an agglomerative approach are its relatively high complexity and that in general it does not guarantee even locally optimal solutions. The multivariate aIB is no exception in this sense. In particular, if we start from $T_j = \mathbf{U}_j$ the time complexity is typically an order of $O(\sum_{j=1}^{k} |\mathcal{U}_j|^3 \mid \mathcal{V}_j|)$, [2] while the space complexity is an order of $O(\sum_{j=1}^{k} |\mathcal{U}_j|^2)$. For the original IB problem we suggested in Section 3.4 a sequential optimization routine to handle these difficulties. In the following we extend this algorithm to solve multivariate IB problems.

Unlike agglomerative clustering, this procedure maintains for each T_j a flat partition with exactly M_j clusters. Given some (e.g., random) initial partitions, at each step we "draw" some $\mathbf{u}_j \in \mathcal{U}_j$ out of its current cluster $t_j(\mathbf{u}_j)$ and represent it as a new singleton cluster. Using our multivariate agglomeration procedure (Eq. (10.8)), we can now merge $\mathbf{u}_j$ into t_j^{new} such that $t_j^{new} = argmin_{t_j \in \mathcal{T}_j} \Delta\mathcal{L}_{max}(\{\mathbf{u}_j\}, t_j)$, to obtain a (possibly new) partition T_j^{new}, with the appropriate cardinality. Assuming that $t_j^{new} \neq t_j(\mathbf{u}_j)$ it is easy to verify that this step increases the value of the functional $\mathcal{L}_{max}$ defined in Eq. (10.2). Since for any finite β this functional is upper bounded, this sequential procedure is guaranteed to converge to a "stable" solution in the sense that no more assignment updates can further improve $\mathcal{L}_{max}$.

In each "draw-and-merge" step we need to calculate the merger costs with respect to each cluster in T_j, which is an order of $O(|\mathcal{T}_j||\mathcal{V}_j|)$. Our time complexity is thus bounded by $O(\ell \cdot \sum_j |\mathcal{U}_j||\mathcal{T}_j||\mathcal{V}_j|)$ where ℓ is the number of iterations we should perform until convergence is attained. Since typically $\ell \cdot |\mathcal{T}_j| \ll |\mathcal{U}_j|^2$ we get a significant run time improvement. Additionally, we improve our memory consumption toward an order of $O(\sum_j |\mathcal{T}_j|^2)$.

As in the case of the multivariate iIB, to reduce the potential sensitivity to local optima, we can repeat this

[2] The notation $|\mathcal{V}_j|$ is loosely used here to refer to the complexity of calculating a single merger in T_j .

```
Input:
    Similar to the multivariate aIB.
    Additional Parameters: Cardinality values $M_j,\ j = 1 : k$ .

Output:
    A partition $T_j$ of $\mathcal{U}_j$ into $M_j$ clusters $\forall j = 1 : k$ .

Initialization:
    For $j = 1 : k,\ \ T_j \leftarrow$ random partition of $\mathcal{U}_j$ into $M_j$ clusters.

Main Loop:
    While not $Done$
        $Done \leftarrow TRUE$ .
        For $j = 1 : k$,
            $\forall\ \mathbf{u}_j \in \mathcal{U}_j$
                Remove $\mathbf{u}_j$ out of $t_j(\mathbf{u}_j)$ .
                $t_j^{new}(\mathbf{u}_j) = \arg\min_{t_j \in \mathcal{T}_j} \Delta\mathcal{L}_{max}(\{\mathbf{u}_j\}, t_j)$ .
                If $t_j^{new}(\mathbf{u}_j) \neq t_j(\mathbf{u}_j)$,
                        $Done \leftarrow FALSE$ .
                Merge $\mathbf{u}_j$ into $t_j^{new}(\mathbf{u}_j)$ .
```

Figure 10.4: Pseudo-code of the multivariate sequential IB algorithm (multivariate sIB). In principle we repeat this procedure for different initializations and choose the solution which maximizes $\mathcal{L}_{max} = \mathcal{I}^{G_{out}} - \beta^{-1}\mathcal{I}^{G_{in}}$.

procedure for different initializations of $\mathbf{T}$ to obtain different solutions, from which we choose the one that maximizes $\mathcal{L}_{max}$. We will term this algorithm *multivariate* sIB. A Pseudo-code is given in Figure 10.4.

Note that in general, different optimization routines are possible to this algorithm. One alternative is to define the outer loop with respect to the $\mathbf{X}$ variables. Here, for every $X_i \in \mathbf{X}$ we iterate over all $x \in \mathcal{X}_i$. For each such value, we perform a "draw-and-merge" step with respect to every T_j for which $X_i \in \mathbf{U}_j$. The second alternative, presented in the figure, is to define the outer loop with respect to the variables in $\mathbf{T}$. Here, for every $T_j \in \mathbf{T}$ we iterate over all $\mathbf{u}_j \in \mathcal{U}_j$, and perform a "draw-and-merge" step for every such value. Other optimization routines are also plausible, and one additional example to the parallel IB case is mentioned in Section 11.1. Obviously different routines will lead to different solutions, and the question of which one is preferable still needs to be explored.

Chapter 11

Multivariate IB Applications

In this chapter we examine a few applications of the general methodology. For brevity, we remain focused on our three running examples: parallel IB, symmetric IB, and triplet IB. For each of these examples we present different applications, using one (ore more) of the algorithmic approaches presented in the previous chapter.

11.1 Parallel IB applications

We consider the specification of G_{in} and $G_{out}^{(a)}$ of Figure 8.2 (upper panel). The relevant distortion measures are given in Eq. (9.4), Eq. (10.11). The complexity of calculating these measures is in principle exponential in k (due to the expectation with respect to $\mathbf{T}^{-j}$). However, while concentrating on "hard" clustering only, one may verify that out of the exponential number of terms, only a few are non zero. Specifically, these terms correspond to the assignments of $\mathbf{T}^{-j}$ such that their disjunction with t_j or with $\bar{t}_j$ are not empty (recall that each assignment of some T_j defines a cluster of X values). Therefore, in this application we concentrate on "hard" clustering solutions only.

Using Eq. (8.6) and Eq. (10.2) we face the problem of *maximizing*

$$\mathcal{L} = I(T_1, \ldots, T_k; Y)) - \beta^{-1}(I(T_1, \ldots, T_k; X) + \mathcal{I}(T_1, \ldots, T_k)) . \qquad (11.1)$$

We further should choose between the multivariate aIB versus sIB algorithms. In this case, the choice is rather simple. Recall that aIB is initialized by singleton clusters, i.e., $T_j = X$ for all $j = 1, \ldots, k$. As a result, at the initial point $|\mathcal{T}| = \Pi_{j=1}^{k} |\mathcal{T}_j| = |\mathcal{X}|^k$, which is extremely redundant. Moreover, it is easy to verify that in this case, the merging criterion of Eq. (10.11) degenerates. Specifically, the JS terms remain equal to zero until we reach a point where $|\mathcal{T}| \approx |\mathcal{X}|$. Only from this point on, we indeed start to compress X (and lose information about Y). [1] To avoid this difficulty we use the sIB approach and choose the initial cardinality values such that $|\mathcal{T}| \ll |\mathcal{X}|$.

As mentioned in Section 10.4 there are different possible optimization routines for the multivariate sIB. Here we describe one alternative which seems suitable in our context. We first perform m sIB restarts with different initializations and $k = 1$, and choose the solution that maximizes Eq. (11.1). [2] We now "freeze" this solution, denoted by T_1, and perform again m sIB iterations with $k = 2$. In other words, *given* T_1, we look for T_2 such that $I(T_1, T_2; Y)) - \beta^{-1}(I(T_1, T_2; X) + \mathcal{I}(T_1, T_2))$ is maximized. We can now "freeze" T_1 and T_2 to look for T_3, and so forth. Loosely speaking, in T_1 we aim to extract the first "principal partition" of the data. In T_2 we seek for the second "principal partition" that is approximately orthogonal (independent) to the first one, and so on.

[1] In fact, this is an inherent problem of the multivariate aIB, which will be eminent for every choice of G_{in} in which some X_i is compressed by more than one of the $\mathbf{T}$ variables.

[2] Note that for $k = 1$ the parallel IB is equivalent to the single-sided IB problem.

Table 11.1: Results for parallel sIB applied to style versus topic text clustering. Each entry indicates the number of "documents" in some cluster and some class. For example, the upper left entry indicates that the first cluster of the first partition, $T_{1,a}$, includes 315 "documents" taken from the book *The Beasts of Tarzan*. The first partition, T_1 is correlated with the writing style, while the second partition, T_2 is correlated with a partition according to the topic.

	$T_{1,a}$	$T_{1,b}$	$T_{2,a}$	$T_{2,b}$
The Beast of Tarzan (Burroughs)	**315**	2	**315**	2
The Gods of Mars (Burroughs)	**407**	0	1	**406**
The Jungle Book (Kipling)	0	**255**	**254**	1
Rewards and Fairies (Kipling)	0	**367**	42	**325**

11.1.1 Parallel sIB for style versus topic text clustering

A well known difficulty in clustering tasks is that there might be more than one meaningful way to partition the data. El-Yaniv and Souroujon [30] mention such a hypothetic example, where a given collection of text documents have two possible dichotomies: by their topics and by their writing styles. Here we construct such an example in practice and solve it using our parallel IB approach.

We selected two authors: E. R. Burroughs and R. Kipling, and downloaded four books from the web site of the Gutenberg Project (http://promo.net/pg/). These are *The Beasts of Tarzan* and *The Gods of Mars* by Burroughs, and *The Jungle Book* and *Rewards and Fairies* by Kipling. Due to this choice, except for the natural partition (according to the writing style), there is indeed an additional possible topic partition (of the "jungle" topic versus all the rest). Our pre-processing included lowering upper case characters, uniting all digits into one symbol and ignoring non alpha-numeric characters. We also ignored the chapter serial titles ("Chapter 1","Chapter 2", etc.) which were present only in the books by Burroughs. We further split each book into "documents" (paragraphs) consisted of 200 (successive) words each, which resulted with $1,346$ documents and $15,528$ distinct words (ignoring the last "short" paragraph in each book). After simple normalization we got an estimated joint distribution $p(d,w)$, where $p(d) = \frac{1}{|\mathcal{D}|}$.

Given these data, we applied the previously described parallel sIB optimization routine to cluster the documents into two clustering hierarchies of two clusters each. Note that this setting implies significant compression, hence we were able to set $\beta^{-1} = 0$. In other words, we simply concentrated on maximizing $I(\mathbf{T};W) = I(T_1,T_2;W)$. Note, though, that even in this case, independent T_j's are preferable to dependent ones (since this independence increases the potential expression power of $\mathbf{T}$). The number of restarts for every T_j was set to be $m = 5$.

In Table 11.1 we present the two different partitions obtained by the algorithm with respect to the different document sources. We see that the first partition, T_1, shows almost perfect correlation to an authorship partitioning. However, the second partition, T_2 is correlated with a topical partitioning, extracting a cluster of mainly "jungle" topic documents, versus a cluster of all the rest. Moreover, $I(T_1;T_2) \approx 0.0007$, i.e., these two partitions are indeed (approximately) independent. Additionally, with only four clusters, $I(T_1,T_2;W) \approx 0.28$ which is about 12.8% of the original information, $I(D;W)$.

We further sorted all words by their contribution to $I(T_1;W)$, given by

$$I(T_1;w) \equiv p(w) \sum_{t_1 \in \mathcal{T}_1} p(t_1 \mid w) \log \frac{p(t_1 \mid w)}{p(t_1)} . \qquad (11.2)$$

In Table 11.2 we present the top five words according to this sort. Clearly for both authors there are different preferences regarding the use of stop words. For example, Burroughs uses the preposition *'of'* more frequently than Kipling, while Kipling uses the verb *'said'* more often than Burroughs (in these specific books). It is reasonable to assume that due to this difference, the first partition, T_1, is correlated with an

Table 11.2: Informative words for the results of parallel sIB, applied to style versus topic text clustering. The left column indicates the word, i.e., W value. The second column specifies its contribution to $I(T_j; W)$. The last four columns indicate the relative frequency of this word in each book (where the larger two values are emphasized). The top five rows are due to a sorting with respect to $I(T_1; W)$. The lower five rows are due to a sorting with respect to $I(T_2; W)$.

W	$I(T_1; w)$	*The Beasts of Tarzan*	*The Gods of Mars*	*The Jungle Book*	*Rewards and Fairies*
'upon'	0.002	**0.008**	**0.006**	0.0005	0.0003
'said'	0.001	0.001	0.002	**0.009**	**0.01**
'of'	0.001	**0.037**	**0.039**	0.024	0.018
'the'	0.001	**0.09**	**0.076**	0.068	0.051
'says'	0.001	0.00005	0.00001	**0.0002**	**0.004**
	$I(T_2; w)$				
'I'	0.002	0.003	**0.025**	0.012	**0.023**
'tarzan'	0.001	0.006	0	0	0
'my'	0.001	0.001	**0.01**	0.004	**0.008**
'jungle'	0.001	**0.003**	0	**0.003**	0
'he'	0.001	0.02	0.006	0.02	0.02

authorship partitioning.

Sorting all words by their contribution to $I(T_2; W)$ (lower rows of Table 11.2), may explain the correlation of the second partition, T_2 with a topical partitioning. Specifically, the word *'jungle'* seems to be a dominant term, "pushing" for this result, as could be expected. However, somewhat unexpectedly, there were additional features, such as *'I'* and *'my'*, supporting this partition.

11.1.2 Parallel sIB for gene expression data analysis

As our second example we used the gene expression measurements of $\sim$ 6800 genes in 72 samples of leukemia [39]. As in many other biological datasets, these data include different (sometimes independent) annotations of their components. Specifically, the sample annotations include type of leukemia (*ALL* vs. *AML*), type of cells, donating hospital, and more.

In our pre-processing we removed $\sim$ 1500 genes that were not expressed in the data and normalized the measurements of the remaining 5288 genes in each sample to get an (estimated) joint distribution $p(s, g)$ over samples and genes (with uniform prior over samples). We sorted all genes by their contribution to $I(S; G)$ (given by $p(g) \sum_s p(s \mid g) \log \frac{p(s|g)}{p(s)}$) and selected the 500 most informative ones (which capture 47% of the original information). After re-normalization of the measurements in each sample we ended up with an estimated joint distribution with $|\mathcal{S}| = 72,\ |\mathcal{G}| = 500$ and $p(s) = \frac{1}{|\mathcal{S}|}$.

Given these data we applied the parallel sIB algorithm to cluster the samples into four clustering hierarchies, with $|\mathcal{T}_j| = 2,\ \forall j = 1:4$. The parameter setting was as in the previous section ($\beta^{-1} = 0,\ m = 5$).

In Table 11.3 we present the four different partitions extracted by the algorithm with respect to different annotations of the data. Note that, again, this comparison is for verification only since these annotations are *not* used during the clustering process which is based on the expression data alone. We see that the first partition, T_1, almost perfectly matches the *AML* vs. *ALL* annotation. The second partition is correlated with the split between B-cells and T-cells (among the samples for which this annotation is provided). For the

Table 11.3: Results for parallel sIB applied to gene expression measurements of leukemia samples [39]. In the upper four rows, each entry indicates the number of samples in some cluster and some class. For example, the upper left entry indicates that the first cluster of the first partition, $T_{1,a}$, includes 23 samples that are all annotated as *AML*. The last row indicates the average PS score of all the cluster samples. Each of the first three partitions is correlated with a different "annotation-dimension" of the samples. Note that T-cell/B-cell annotations are available only for samples annotated as ALL type.

	$T_{1,a}$	$T_{1,b}$	$T_{2,a}$	$T_{2,b}$	$T_{3,a}$	$T_{3,b}$	$T_{4,a}$	$T_{4,b}$
AML	**23**	**2**	14	11	12	13	13	12
ALL	**0**	**47**	37	10	9	38	22	25
B-cell	0	38	**37**	**1**	6	32	20	18
T-cell	0	9	**0**	**9**	3	6	2	7
average PS	0.64	0.72	0.71	0.66	**0.53**	**0.76**	0.70	0.69

third partition we note that the average "Prediction Strength" (PS) score [3] is very different between both clusters. In particular, in the first cluster, only 3 samples (out of 21) had a PS score greater than 0.75 while in the second cluster, 34 samples (out of 51) exceed this 0.75 threshold.

For the fourth partition we were not able to find any clear correlation with one of the available annotations. This raises the possibility that while extracting this partition the algorithm in fact over fits the data. In terms of information, $I(\mathbf{T};G)$ preserves almost 54% of the original information, $I(S;G) \approx 0.23$.

11.2 Symmetric IB applications

We consider the specification of G_{in} and $G_{out}^{(a)}$ of Figure 8.2 (middle panel) and the alternative variational principle (Eq. (8.3)). As already mentioned (Example 9.2.2), equivalently we may drop the edges between T_X and X and between T_Y and Y (in G_{out}), and use the first variational principle (Eq. (8.2)). Either way we face the problem of *minimizing*

$$\mathcal{L} = I(T_X;X) + I(T_Y;Y) - \beta I(T_X;T_Y) \ , \tag{11.3}$$

or *maximizing*

$$\mathcal{L} = I(T_X;T_Y) - \beta^{-1}(I(T_X;X) + I(T_Y;Y)) \ . \tag{11.4}$$

The relevant distortion measures are given in Eq. (9.5) and Eq. (10.12). In contrast to the previous parallel IB example, here there are no major complexity difficulties. Therefore, we are able to apply all the different algorithmic approaches as we demonstrate in the following.

11.2.1 Symmetric dIB and iIB for word-topic clustering

We start with a simple text processing example. We use the same subset of the 20NG corpus, already described in Section 4.3. Specifically, this subset is represented as an estimated joint distribution $p(w,c)$ of 200 "informative" words versus 20 (topical) categories. That is, $|\mathcal{W}| = 200$, $|\mathcal{C}| = 20$, and each entry indicates the estimated probability that a random word position is equal to $w \in \mathcal{W}$ while at the same time the topic of its document is $c \in \mathcal{C}$.

[3] The Prediction Strength (PS) score was defined in [39] as an estimate (between 0 to 1) of how well one may predict the type of leukemia, based on the expression levels of a fixed subset of genes. This subset was chosen based on their expression correlation with the class distinction in the initial ("training") set of 38 samples. See [39] for the details.

Given these data we applied the symmetric dIB algorithm (with asynchronous updates) to cluster *both dimensions* into two sets of clusters: clusters of words, T_w, and clusters of categories, T_c. The implementation details were similar to those mentioned in Section 4.3 for applying the single-sided dIB to these data. Specifically the rate of increasing β was defined through $f_\beta = (1 + \varepsilon_\beta)\beta,\ \varepsilon_\beta = 0.001$. The parameters used for detecting splits were defined by $d_{min} = \frac{1}{\beta}$ (for both T_X and T_Y), i.e., as β increases the algorithm becomes more "liberal" for declaring cluster splits. The scaling factor for the stochastic duplication was set to $\alpha = 0.005$. [4]

The partition of $\mathcal{C}$ induced by T_C was typically "hard", i.e., for every $c \in \mathcal{C}$, $p(t_c \mid c)$ was approximately 1 for one cluster and 0 for all the others. Hence, we are able to present the hierarchy found by T_c as a simple tree-like structure, given in Figure 11.1. This hierarchy is in high agreement with a topical partitioning of the categories. However, it is not really a tree. Specifically, the *electronics* category is assigned to the left branch after the first split (or phase transition). After the next split it is assigned to a cluster in the right branch. Finally, after another split, it is assigned back to a cluster in the left branch.

Considering the word clusters of the T_w hierarchy (after four splits) we see that each of these clusters is correlated with one of the clusters in T_c. To demonstrate this we find for every $t_c \in \mathcal{T}_c$ its most probable word cluster, given by $t_w^* = \mathrm{argmax}_{t_w} p(t_w \mid t_c)$. For this cluster we present in Figure 11.1 its five most probable words, i.e., the five words that maximize $p(w \mid t_w^*)$. Clearly, these words show high semantic agreement with the topics of the relevant category cluster. The mapping of W values into T_w also utilized the "soft" clustering utility, which is available by the algorithm to deal with words that are relevant to several category clusters. Thus, some of the words were assigned to more than one cluster. For example, the word *'war'* was assigned with high probability to a cluster t_w, for which the most probable category cluster, t_c^*, was the "politics" cluster. Additionally, it was assigned with lower probability to two other word clusters for which t_c^* was the "religion-mideast" cluster. Other examples are given in Table 11.4 (which is, naturally, somewhat reminiscent of Table 4.2).

In terms of information, after four splits, we get $|\mathcal{T}_w| = 14,\ |\mathcal{T}_c| = 9$ and $I(T_w; T_c) = 0.59$, which is about 71% of the original information, $I(W; C)$. In other words, although the number of entries in the joint distribution matrix $p(t_w, t_c)$ is only 3% with respect to the number of entries in $p(w, c)$, most of the information is preserved.

We further applied the symmetric iIB algorithm to the same data. For purposes of comparison, the input parameters were set by using the dIB result. Specifically we set $|\mathcal{T}_w| = 14,\ |\mathcal{T}_c| = 9$ and $\beta \approx 22.72$ which corresponds to the β value right after the fourth split in dIB. We performed 100 different random ("hard") initializations and used $\varepsilon = 0.001$ to declare convergence. In Figure 11.2 we see that only 8 of these 100 restarts converged to a better minimum of $\mathcal{L}$ than the one found by dIB. In particular, in these 8 cases (and also in another single case), the iIB solution attained (slightly) higher $I(T_w; T_c)$ than the dIB solution. These results highlight several issues. First, even for a relatively simple problem, many different locally optimal solutions are present (as we already saw for the original single-sided IB problem). Second, by "tracking" the changes in the solution, starting at the simple case of two clusters in each hierarchy, dIB succeeds in finding a relatively good solution. Moreover, it provides more details by describing a hierarchy of solutions in different resolutions. Nonetheless, if one is interested in a "flat" solution for a given number of clusters, using iIB with a sufficient number of initializations will probably provide a better optimum.

Considering T_c for the best iIB solution (in terms of $\mathcal{L}$), we see that although it is different from the dIB solution, both solutions are strongly correlated. For example, the three "religion" categories are still in a single cluster, and so are the two "sport" categories, the two "hardware" categories and the autos and motorcycles categories. Interestingly, the ambiguity regarding the "electronics" category also remains in this

[4] It is easy to verify that $p(t_j \mid \mathbf{u}_j) = p(t_j)$ is a (trivial) fixed point of the symmetric IB-functional that we are trying to optimize, *for any value of* β. On the other hand, if α is small, we initialize the first two copies in T_X by $p(t_X \mid x) \approx p(t_X) \approx 0.5$ (and similarly for T_Y). Thus, before the first split, the initialization of the duplicated copies in T_X and T_Y is typically very close to this trivial fixed-point. To avoid this attractor one must use a larger α value, and we chose $\alpha_1 = 0.95$. Note that this value is utilized only before the first split.

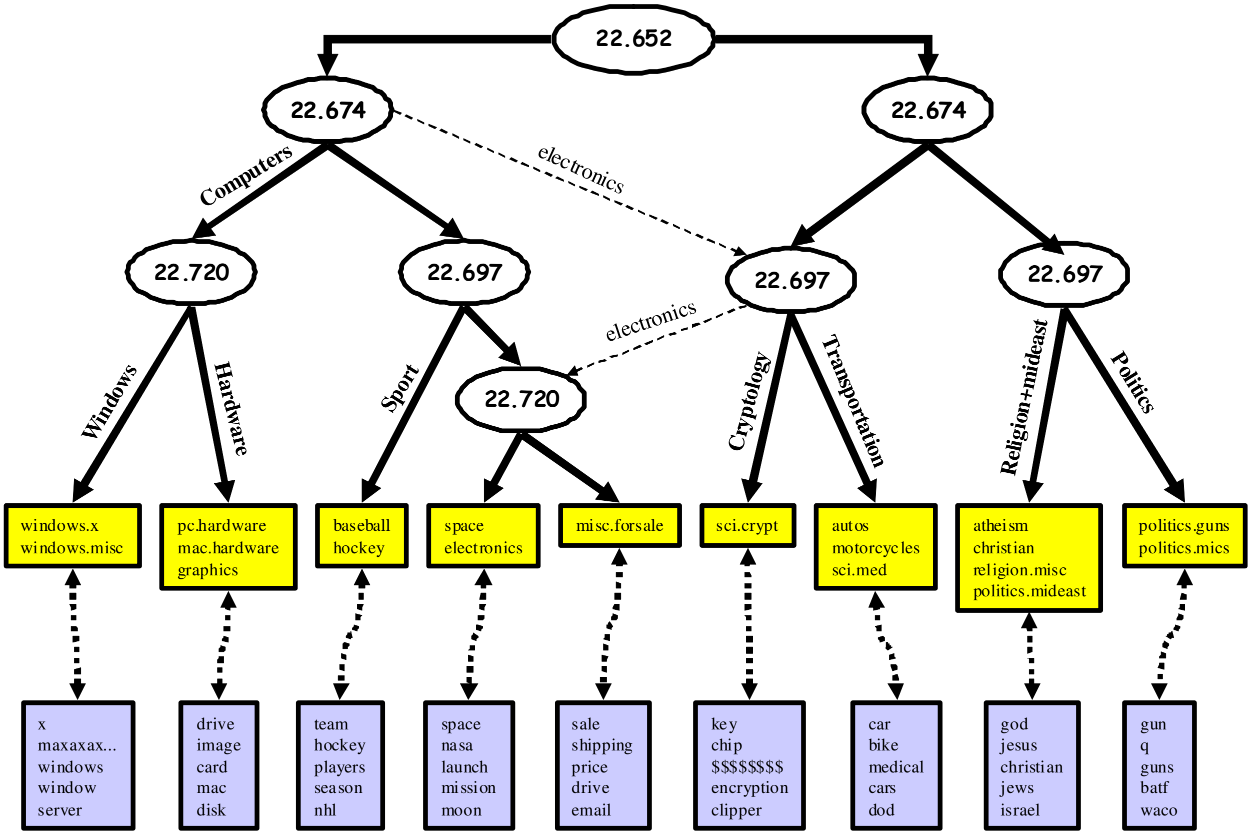

Figure 11.1: Application of the Symmetric dIB to the $20NG$ word-category data. The learned cluster hierarchy of categories, T_c, is presented after four phase transitions. The numerical value inside each ellipse denotes the value of β for which the corresponding cluster bifurcated. In general, this hierarchy is in high agreement with a topical partitioning of the categories. In the lower level, for each $t_c \in \mathcal{T}_c$ we find the most probable cluster of words (defined as $t^*_w = argmax_{t_w \in \mathcal{T}_w} p(t_w \mid t_c)$). Given this cluster we sort all words by $p(w \mid t^*_w)$ and present the top five words. As can be seen from the figure, these words are well correlated with the corresponding category cluster (the character '\$' stands for a digit character). Also note that at the early stages, the algorithm is inconclusive regarding the assignment of the *electronics* category. After the first split it is in the left branch of the tree. After the next bifurcation it is assigned to a cluster in the right branch, and after another phase transition, it is returned to the left branch. This phenomenon demonstrates that the hierarchy obtained by the dIB algorithm does not necessarily construct a tree.

Table 11.4: Results for "soft" word clustering using symmetric dIB over the 20NG word-category data. The first column indicates the word, i.e., W value. The second column presents $p(t_w \mid w)$ for all the different clusters $t_w \in \mathcal{T}_w$ for which this probability was non zero. Given each t_w, t_c^* denotes the most probable category cluster, i.e., the category cluster that maximizes $p(t_c \mid t_w)$. It is represented in the table (in the third column) by the joint topic of its members, which are the categories for which $p(t_c^* \mid c) \approx 1$ (see Figure 11.1). The last column presents the probability of this cluster given t_w.

W	$p(t_w \mid w)$	t_c^*	$p(t_c^* \mid t_w)$
war	0.92	*politics*	0.44
	0.06	*religion-mideast*	0.34
	0.02	*religion-mideast*	0.93
killed	0.86	*politics*	0.44
	0.08	*religion-mideast*	0.34
	0.06	*religion-mideast*	0.93
evidence	0.77	*religion-mideast*	0.34
	0.23	*politics*	0.44
price	0.74	*hardware*	0.31
	0.26	*sport*	0.35
speed	0.99	*hardware*	0.31
	0.01	*sport*	0.35
application	0.58	*hardware*	0.31
	0.42	*windows*	0.84

Table 11.5: Dataset details of the protein GST domain test.

class	**family name**	#proteins
c_1	*GST* - no class label	298
c_2	*S crystallin*	29
c_3	*Alpha class GST*	40
c_4	*Mue class GST*	32
c_5	*Pi class GST*	22

iIB solution. Specifically, it is assigned with probability 0.9 to the "hardware" cluster and with probability 0.1 to the autos and motorcycles cluster.

11.2.2 Symmetric sIB and aIB for protein clustering

As a second example we used a subset of five protein classes taken from the PRINTS database [4] (see Table 11.5 for details). [5] These data were already used in [72] to examine the effectiveness of *supervised* classification techniques. All five classes share a common domain (a domain is an independent protein structural unit), known as the glutathione S-transferase (GST) domain. We specifically chose this test since a well established database of protein families HMMs, [6] currently considered the state-of-the-art in generative modeling of protein families, has chosen not to model these groups separately, due to high sequence similarity between members of the different groups. In spite of this potential pitfall, we find that *unsupervised* clustering using symmetric IB algorithms may extract clusters that are well correlated with the

[5] The author is grateful to Gill Bejerano for preparing these data and for his help in analyzing the results presented in this section.

[6] The Pfam database, http://www.sanger.ac.uk/Pfam.

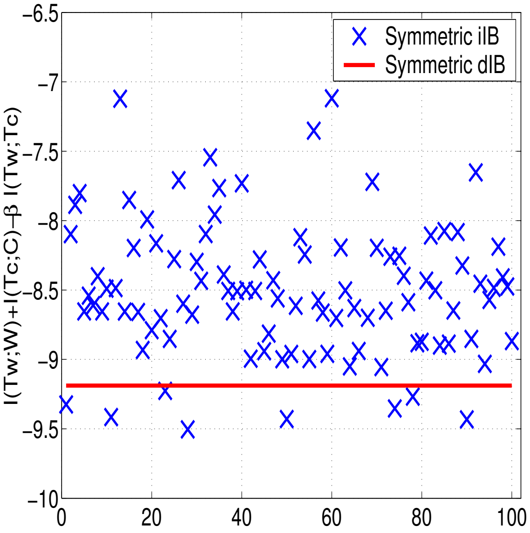

Figure 11.2: Application of symmetric dIB and symmetric iIB to the 20NG word-category data. In 8 out of 100 different initializations, iIB converges to a better minimum of $\mathcal{L} = I(T_w; W) + I(T_c; C) - 22.72 \cdot I(T_w; T_c)$.

different groups.

For these data, obviously there is no clear definition of "words", and usually each protein is described by its ordered sequence of amino-acids. Our pre-processing included representing each protein as a counts vector with respect to all the different 4-grams of amino-acids present in these data. Denoting this set of features by $\mathcal{F}$ and the set of proteins by $\mathcal{R}$, we got a counts matrix of $|\mathcal{R}| = 421$ proteins versus $|\mathcal{F}| = 38,421$ features. After normalizing the counts for each protein to unity we got a joint distribution $p(r, f)$ with $p(r) = \frac{1}{|\mathcal{R}|}$. To avoid overly high dimensionality, we sorted all features by their contribution to $I(F; R)$ and selected the top $2,000$ (which capture about 22% of the original information). After re-normalization we ended up with a joint distribution $p(r, f)$ with $|\mathcal{R}| = 421, \ |\mathcal{F}| = 2,000$ and $p(r) = \frac{1}{|\mathcal{R}|}$.

As in the previous text example, we apply two algorithms to these data. The symmetric aIB provides a (tree structure) hierarchy of solutions at all the different resolutions. In contrast, the symmetric sIB provides m different "flat" solutions (at a given resolution), from which the best one should be used. As in Section 11.1.1 we set $\beta^{-1} = 0$, hence we were interested in extracting clusters of proteins, T_R, and clusters of 4-grams of amino-acids, T_F, such that $\mathcal{L} = I(T_R; T_F)$ is maximized. We start by describing the results obtained by the symmetric sIB, for $|\mathcal{T}_R| = 10, \ |\mathcal{T}_F| = 10$.

One issue we should address while using the symmetric sIB is how to initialize T_R and T_F. Consider the relevant distortion measure given in Eq. (10.12). Random initialization of both T_R and T_F is clearly problematic since the mergers in T_R and T_F will initially take place based on an effectively random joint distribution $p(t_R, t_F)$. A possible solution is to use the strategy described in [77]. Specifically, we randomly initialize only T_F and optimize it using the original *single-sided* sIB algorithm, such that $I(T_F; R)$ is maximized. The obtained set of clusters provides a robust low-dimensional representation for the proteins. Given this representation we randomly initialize T_R and use again the original *single-sided* sIB algorithm to optimize it such that $I(T_R; T_F)$ is maximized. We use these two solutions as the initialization to the *symmetric* sIB algorithm, and continue by the general framework described in Figure 10.4 until convergence is attained. We repeat this procedure 100 different times and select the best solution, i.e., the one which maximizes $I(T_R; T_F)$.

For this solution we find that with only ten clusters of proteins and ten clusters of features, $I(T_R; T_F) = 1.06$ which is about 30% of the original information. In Table 11.6 we see that the correlation of the

Table 11.6: Results for applying symmetric sIB to the GST protein dataset with $|\mathcal{T}_R| = 10,\ |\mathcal{T}_F| = 10$. Each entry indicates the number of proteins in some cluster and some class. For example, the upper left entry indicates that the first cluster t_{R_1}, includes 107 proteins, all of them from the ("unlabeled") class c_1. All the ten protein clusters are well correlated with the (biological) partition of the proteins into classes. The last row indicates the number of "errors" for each cluster, defined as the number of proteins in this cluster, which are *not* labeled by the cluster's most dominant label. Overall, there are 17 errors among the 421 proteins, i.e., the correlation (or the micro-averaged precision) is 96%.

class/cluster	t_{R_1}	t_{R_2}	t_{R_3}	t_{R_4}	t_{R_5}	t_{R_6}	t_{R_7}	t_{R_8}	t_{R_9}	$t_{R_{10}}$
c_1	**107**	**49**	**47**	**42**	**30**	**17**	4	1	1	0
c_2	0	0	0	0	0	0	**29**	0	0	0
c_3	0	0	0	0	0	0	0	**39**	0	1
c_4	0	0	0	0	0	2	0	0	**30**	0
c_5	0	7	0	0	0	0	0	0	1	**14**
Errors	0	7	0	0	0	2	4	1	2	1

protein clusters with the available class labels is almost perfect. Hence, the algorithm recovers the (manual) biological partitioning of the proteins into classes.

As in the previous text example, we further analyze the feature clusters given in T_F. First we identify for each $t_R \in \mathcal{T}_R$ its most probable feature cluster, defined as $t_F^* = \mathrm{argmax}_{t_F} p(t_F \mid t_R)$. For this cluster we present in Table 11.7 its three most probable features, i.e., the three 4-grams that maximize $p(f \mid t_F^*)$. Examining the relative frequencies of these features in the different classes, we see that almost all of them are good indicators for the biological class that is correlated with the protein cluster, t_R.

To summarize, although our analysis is entirely unsupervised, we are able to extract clusters that are correlated with a manual partitioning of the proteins into biologically meaningful classes. Moreover, we identify features (4-grams) that seem to be good indicators for each such class. Further analysis of these results, including the possible biological functionality of the features presented in Table 11.7, will be presented elsewhere.

Lastly, we apply the symmetric aIB to the same data. Recall that this algorithm extracts solutions in all the different resolutions. For purposes of comparison, we first consider the solution at $|\mathcal{T}_R| = 10,\ |\mathcal{T}_F| = 10$. In terms of information, this result is clearly inferior to the symmetric sIB result. Specifically, using aIB we have $I(T_R; T_F) = 0.93$ which is about 26.5% of the original information. Moreover, *all* the 100 different solutions obtained by symmetric sIB attained higher information values, which demonstrates the fact that the aIB approach is not guaranteed to converge even to a locally optimal solution. However, a more careful examination of these results shows that the differences are mainly in the T_F partition. In other words, the aIB T_R solution is strongly correlated with the corresponding sIB solution (and thus, also well correlated with the protein class labels). Therefore we conclude that the T_F solution found by aIB is sub-optimal, which leads to the overall inferiority in terms of information.

Nonetheless, one of the advantages of the aIB algorithm is that one may consider a hierarchy of solutions. In Figure 11.3 we present this tree structured hierarchy (for T_R). We see that one branch of the tree contains mostly proteins from the classes $c_2,\ c_3$ and c_4. The other branch consists of almost all the "unlabeled" (c_1 class) proteins, accompanied by the *Pi class* proteins (class c_5). Since we have several clusters that correspond to the "unlabeled" c_1 class, we suggest that these clusters are correlated with additional, yet unknown sub classes in the *GST* domain.

In fact, after completing our experiments it was brought to our attention that one such new class was recently defined in a different database, the InterPro database [2]. Out of the 95 proteins available for this

Table 11.7: Results for symmetric sIB: Indicative features for GST protein classes. The left column indicates the index of the cluster in $\mathcal{T}_R$ (indices are the same as in Table 11.6) and the most dominant class in this cluster. The second column indicates the index of the cluster in $\mathcal{T}_F$ for which $p(t_F \mid t_R)$ is maximized (denoted by t_F^*). The next column indicates this maximizing value, i.e., $p(t_F^* \mid t_R)$. Results are presented only when this value is greater than 0.8, indicating high coupling between the feature cluster and the protein cluster. We further sort all features by $p(f \mid t_F^*)$ and present the top three features in the next column. The last five columns indicate for each of these features, its relative frequency in all five classes (estimated as the number of occurrences of this feature in proteins from the class, divided by the total number of occurrences of all features in this class). As can be seen in the table, the extracted features are correlated with the biological class associated with the protein cluster, t_R. Recall that these results are based on an entirely unsupervised analysis.

T_R value	t_F^*	$p(t_F^* \mid t_R)$	Feature	$p(f \mid t_F^*)$	c_1	c_2	c_3	c_4	c_5
t_{R_7} (c_2)	$t_{F_{10}}$	0.91	*RYLA*	0.022	0.002	**0.009**	0	0.002	0
			GRGR	0.020	0.001	**0.006**	0.005	0	0
			NGRG	0.019	0.001	**0.006**	0.003	0	0
t_{R_9} (c_4)	t_{F_8}	0.89	*FPNL*	0.025	0.001	0	0	**0.014**	0
			AILR	0.018	0.001	0	0	**0.007**	0.006
			SNAI	0.017	0.001	0	0	**0.008**	0.004
$t_{R_{10}}$ (c_5)	t_{F_2}	0.85	*LDLL*	0.019	0.001	0	0.001	0	**0.006**
			SFAD	0.017	0.001	0	0	0	**0.006**
			FETL	0.017	0.001	0	0	0	**0.006**
t_{R_8} (c_3)	t_{F_1}	0.85	*FPLL*	0.018	0	0	0.007	0	**0.008**
			YGKD	0.017	0.001	0	**0.007**	0	0.005
			AAGV	0.016	0.001	0	**0.006**	0	0
t_{R_5} (c_1)	t_{F_9}	0.83	*TLVD*	0.015	**0.003**	0	0	0	0
			WESR	0.015	**0.003**	0	0	0	0
			EFLK	0.015	**0.002**	0	0	0.002	0
t_{R_4} (c_1)	t_{F_5}	0.80	*IPVL*	0.010	**0.002**	0	0	0	0
			ARFW	0.010	**0.002**	0	0	0	0
			KIPV	0.009	**0.002**	0	0	0	0

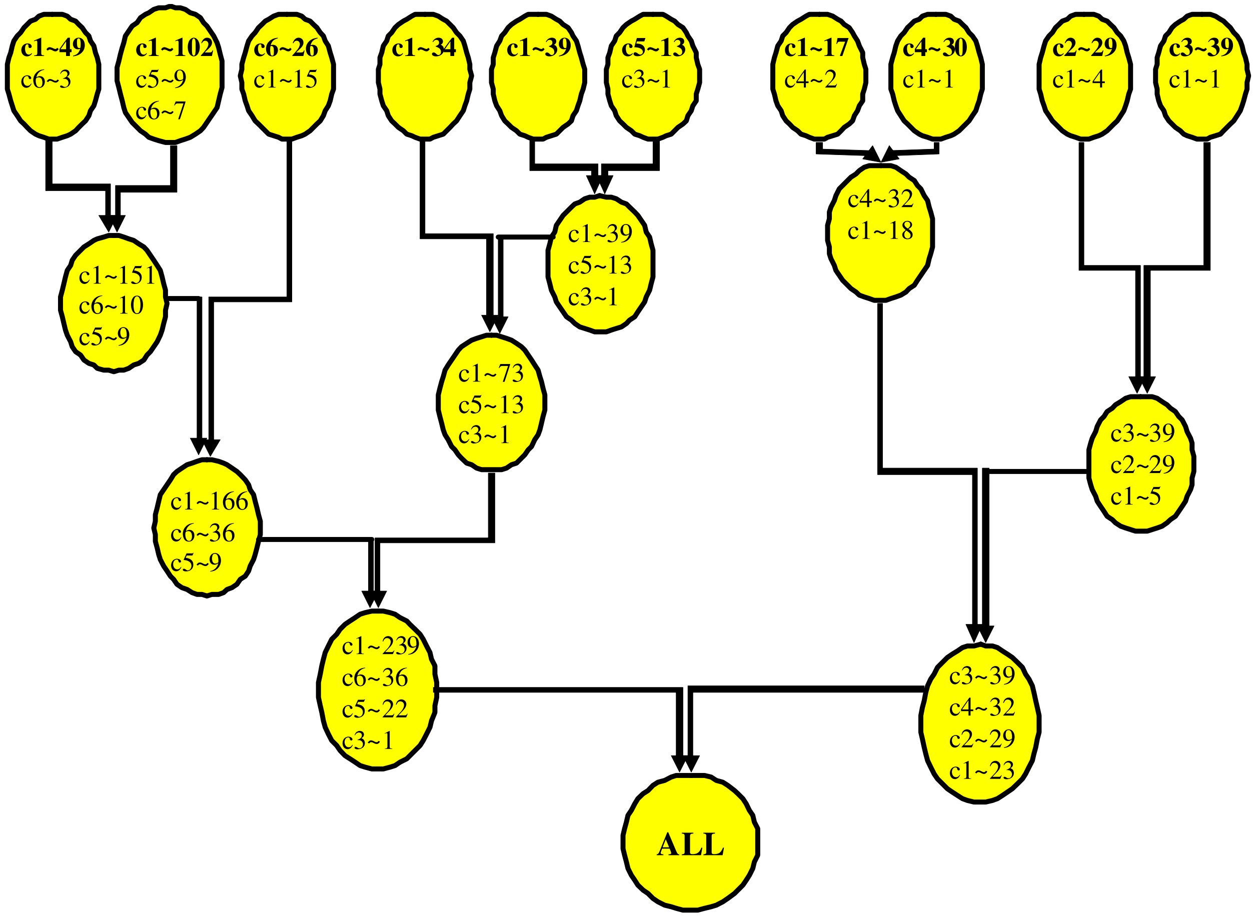

Figure 11.3: Application of the symmetric aIB to the GST protein dataset. The learned protein cluster hierarchy, T_R is presented from $|\mathcal{T}_R| = 10$ and below. In each cluster the number of proteins from every class is indicated. For example, in the extreme right (upper) cluster there are 39 proteins from the class c_3 and a single protein from the unlabeled class c_1. In general, the right branches correspond to proteins from the classes c_2, c_3 and c_4. The left branches correspond to proteins from the class c_5 and from the unlabeled class, c_1. After completing the experiments we found out that 36 of the proteins in this class were recently labeled as *Omega* class. This class is denoted by c_6 in the figure. Note that all its proteins were clustered in the three left-most clusters.

new (*Omega*) class, 36 were present in our data (labeled as c_1). [7] In Figure 11.3 we see that these 36 proteins are present in three ("c_1") clusters, which are all merged together in a later stage (with no additional clusters). Note especially that one of these clusters consists of 26 *Omega* proteins and 15 unlabeled C_1 proteins. This suggests that at least some of these 15 proteins will also be identified as *Omega* class proteins in the future.

11.3 Triplet IB application

We conclude this chapter with a simple application of the triplet IB in the context of natural language modeling. We consider the specification of G_{in} and $G^{(a)}_{out}$ of Figure 8.2 (lower panel) and the first variational

[7] Currently this class is also defined in the PRINTS database [4]. However, since the 16 proteins available for it were included among the 95 available from InterPro, we used the InterPro data for this class.

principle given in Eq. (8.2). As already mentioned in Section 8.4.3, equivalently we may consider the specification $G^{(b)}_{out}$ in the same figure and the alternative variational principle, Eq. (8.3). In both cases we face the problem of maximizing

$$\mathcal{L} = I(T_p, T_n; Y) - \beta^{-1}(I(T_p; X_p) + I(T_n; X_n)) \ . \tag{11.5}$$

Due to similar complexity considerations as those mentioned in Section 11.1, the natural and most simple choice in this case is to use the sIB algorithm. Using Theorem 10.3.3 we find that the relevant distortion measure is given by

$$\Delta\mathcal{L}(t_p^\ell, t_p^r) = p(\bar{t}_p) \cdot (E_{p(\cdot|\bar{t}_p)}[JS_{\Pi_{t_n}}[p(y \mid t_n, t_p^\ell), p(y \mid t_n, t_p^r)]] - \beta^{-1}H(\Pi)) \ , \tag{11.6}$$

and an analogous expression for mergers in T_n.

11.3.1 Triplet sIB for natural language processing

To collect the input joint statistics we used the seven Tarzan books by E. R. Burroughs, available from the Gutenberg project. These are *Tarzan and the Jewels of Opar*, *Tarzan of the Apes*, *Tarzan the Terrible*, *Tarzan the Untamed*, *The Beasts of Tarzan*, *The Jungle Tales of Tarzan*, and *The Return of Tarzan*. We followed the same pre-processing steps as in Section 11.1.1, ending up with a sequence of $580,806$ words taken from a vocabulary of $19,458$ distinct words. We defined three random variables, corresponding to the previous, current and the next word in the sequence. We denote these variables by W_p, W, and W_n, respectively. To avoid complexity difficulties we defined W to be the set of ten most frequent words in the above books, which are *not* stop-words. Specifically, these were *'apemans', 'apes', 'eyes', 'girl', 'great', 'jungle' 'tarzan', 'time', 'two'* and *'way'*. Hence, we considered word triplets in which the middle word was one of these ten words. After ignoring triplets with less than three occurrences, we had 672 different triplets with a total of $4,479$ occurrences. In these triplets, the number of distinct first-words was 90 and the number of distinct last-words was 233. Thus, after simple normalization we had an estimated joint distribution $p(w_p, w, w_n)$ with $|\mathcal{W}_p| = 90,\ |\mathcal{W}| = 10,\ |\mathcal{W}_n| = 233$.

Given these data we applied the triplet sIB algorithm to construct two systems of clusters: T_p for the first-word in the triplets, and T_n for the last-word in the triplets. We set $|\mathcal{T}_p| = 10,\ |\mathcal{T}_n| = 10$, and since this setting already implies significant compression we were able to take $\beta^{-1} = 0$ and simply concentrate on maximizing $I(T_p, T_n; W)$. As in the symmetric IB case, a direct random initialization of both T_p and T_n might be problematic since in this case the first mergers will take place based on an effectively random joint distribution (see Eq. (11.6)). Hence, we randomly initialize T_p and optimize it using the original *single-sided* sIB algorithm, such that $I(T_p; W)$ is maximized. Similarly, we initialize T_n such that $I(T_n; W)$ is maximized. Using these initializations and the general scheme described in Figure 10.4, we optimize both systems of clusters until they converge to a local maximum of $I(T_p, T_n; W)$. We repeat this procedure for 50 different initializations to extract different locally optimal solutions.

In terms of information, each of these 50 solutions preserved more than 91% of the original information, $I(W_p, W_n; W) = 1.63$. This result is of special interest, taking into account that the dimensions of the joint distribution $p(t_p, w, t_n)$ are more than 200 times smaller than those of the original matrix, $p(w_p, w, w_n)$. The best solution preserved about 93.5% of the original information and we further concentrate on this solution.

In Table 11.8 we present for every $w \in \mathcal{W}$, the couple of clusters, t_p^*, t_n^* for which $p(t_p, w, t_n)$ is maximized. For each such couple we sort all members, $w_p \in t_p^*,\ w_n \in t_n^*$ by $p(w \mid w_p, w_n)$ and present the top four pairs. In many cases these pairs are indicative of the "in-between" word, w, which reflects how T_p and T_n preserve the information about W.

We further validate the predictive power of T_p and T_n about W by the following experiment, in which we scan another book by E. R. Burroughs (again, taken from the Gutenberg project), which is *The Son of Tarzan*. Note that this book was *not* used during our "training", where we estimated $p(w_p, w, w_n)$ and extracted T_p

Table 11.8: Results for triplet sIB. In the left column we present the word $w \in \mathcal{W}$. The next two columns indicate the couple of clusters, $t_p^* \in \mathcal{T}_p,\ t_n^* \in \mathcal{T}_n$ for which $p(t_p, w, t_n)$ is maximized. This maximizing value is presented in the fourth column. The next three columns indicate the four pairs of words, $w_p \in t_p^*,\ w_n \in t_n^*$ for which $p(w \mid w_p, w_n)$ is maximized, where the middle word is repeated here for convenience (ties are solved by a further sorting with respect to $p(w_p, w_n)$). The last column presents the probability of the middle word given these pairs.

W value	t_p^*	t_n^*	$p(t_p^*, w, t_n^*)$	W_p value	W value	W_n value	$p(w \mid w_p, w_n)$
apeman	t_{p_3}	t_{n_3}	0.05	*the*	*apeman*	*leaped*	0.67
				the	*apeman*	*knew*	0.64
				the	*apeman*	*took*	0.63
				the	*apeman*	*realized*	0.62
apes	t_{p_3}	t_{n_3}	0.03	*the*	*apes*	*mighty*	0.63
				the	*apes*	*became*	0.50
				the	*apes*	*did*	0.50
				the	*apes*	*sat*	0.44
eyes	t_{p_6}	t_{n_3}	0.02	*his*	*eyes*	*were*	1.00
				his	*eyes*	*wandered*	1.00
				his	*eyes*	*had*	1.00
				his	*eyes*	*narrowed*	1.00
girl	t_{p_3}	t_{n_6}	0.02	*the*	*girl*	*shuddered*	1.00
				the	*girl*	*cast*	1.00
				the	*girl*	*heard*	1.00
				the	*girl*	*asked*	1.00
great	t_{p_3}	t_{n_2}	0.07	*the*	*great*	*apes*	1.00
				the	*great*	*beast*	1.00
				the	*great*	*ape*	1.00
				the	*great*	*cat*	1.00
jungle	t_{p_3}	t_{n_7}	0.03	*the*	*jungle*	*before*	0.73
				the	*jungle*	*his*	0.63
				the	*jungle*	*there*	0.63
				the	*jungle*	*as*	0.50
tarzan	t_{p_8}	t_{n_3}	0.04	*which*	*tarzan*	*had*	1.00
				as	*tarzan*	*had*	1.00
				which	*tarzan*	*was*	1.00
				but	*tarzan*	*was*	1.00
time	t_{p_5}	$t_{n_1}0$	0.02	*this*	*time*	*he*	1.00
				this	*time*	*the*	1.00
				long	*time*	*he*	1.00
				same	*time*	*he*	1.00
two	t_{p_3}	t_{n_1}	0.02	*the*	*two*	*men*	1.00
				the	*two*	*priests*	1.00
				the	*two*	*approached*	1.00
				the	*two*	*lay*	1.00
way	t_{p_6}	t_{n_5}	0.01	*his*	*way*	*with*	1.00
				his	*way*	*toward*	0.77
				his	*way*	*to*	0.73
				her	*way*	*to*	0.33

and T_n. For every occurrence in this book of one of the ten words in $\mathcal{W}$, we try to predict it using its two immediate neighbors, in several different ways. Let w_p and w_n be the previous and next word, before and after a word w, respectively. If these two words occurred in our training data, [8] i.e., $w_p \in \mathcal{W}_p,\ w_n \in \mathcal{W}_n$ then their assignments in $T_p,\ T_n$ define a specific couple of clusters, $t_p \in \mathcal{T}_p,\ t_n \in \mathcal{T}_n$. Given this couple, we predict the in-between word to be $\hat{w} = \mathrm{argmax}_w p(w \mid t_p, t_n)$. Given these predictions, for every $w \in \mathcal{W}$ we can calculate the following quantities: $A_1(w)$ defines the number of w occurrences correctly predicted as w (true-positives), $A_2(w)$ defines the number of words incorrectly predicted as w (false-positives), and $A_3(w)$ defines the number of w occurrences incorrectly not predicted as w (false-negatives). The precision and recall for w is then defined as $Prec(w) = \frac{A_1(w)}{A_1(w)+A_2(w)}$, $Rec(w) = \frac{A_1(w)}{A_1(w)+A_3(w)}$, where the micro-averaged precision and recall are defined by (see Section 4.5.2)

$$< Prec >= \frac{\sum_w A_1(w)}{\sum_w A_1(w) + A_2(w)} \ , < Rec >= \frac{\sum_w A_1(w)}{\sum_w A_1(w) + A_3(w)} \ . \tag{11.7}$$

for purposes of comparison we applied two additional prediction schemes. The first uses the original joint statistics, $p(w_p, w, w_n)$, estimated by the training data. Namely, given w_p and w_n we predict the in-between word to be $\hat{w} = \mathrm{argmax}_w p(w \mid w_p, w_n)$. The second and third use just one neighbor for the prediction. Namely, given w_p we predict the next word to be $\hat{w} = \mathrm{argmax}_w p(w \mid w_p)$, and given w_n we predict the previous word to be $\hat{w} = \mathrm{argmax}_w p(w \mid w_n)$. In Table 11.9 we present the precision and recall for all the ten words in W, using all the above mentioned prediction schemes. Interestingly, in spite of the significant compression implied by T_p and T_n, the (averaged) precision of its predictions is similar to those obtained using the original complete joint statistics. Moreover, in terms of recall, predictions that use the triplet IB clusters are (on the average) superior to those using the original $W_p,\ W_n$ variables. This is probably due to the fact that while using $p(w_p, w, w_n)$ to predict the in-between word in a specific triplet, this specific triplet must occur in the training data. On the other hand, while using $p(t_p, w, t_n)$ a prediction can be provided even for new triplets, for which only their individual components occurred in the training data. Lastly, we observe that using both word neighbors, instead of using only the previous or next word, significantly improves the precision of the predictions.

It should be noted that in principle this type of application might be useful in tasks like speech recognition, optical character recognition and more. Moreover, for these tasks typically it is not feasible to use the original joint distribution due to its high dimensionality. Using the triplet IB clusters might be a reasonable alternative in these situations, which is dramatically less demanding. Additionally, for biological sequence data, the analysis demonstrated in this section might be useful to gain further insights about the data properties.

[8] Note that this is not necessarily true, since we test over a *new* sequence. In these cases no prediction is provided.

Table 11.9: Precision and Recall results for triplet sIB. The left column indicates the word $w \in \mathcal{W}$ and in parentheses its number of occurrences in the test sequence. The next column presents the precision of the predictions while using the triplet sIB clusters statistics, i.e., $p(w \mid t_p, t_n)$. The third column presents the precision while using the original joint statistics, i.e., $p(w \mid w_p, w_n)$. Note that this joint distribution matrix is about 200 times larger than the previous one. The next two columns present the precision while using only one word neighbor for the prediction, i.e., $p(w \mid w_p)$ and $p(w \mid w_n)$, respectively. The last four columns indicate the recall of the predictions while using these four different prediction schemes. The last row presents the micro-averaged precision and recall.

	Precision				**Recall**			
W	T_p, T_n	W_p, W_n	W_p	W_n	T_p, T_n	W_p, W_n	W_p	W_n
apeman (33)	5.9	7.4	4.3	1.5	24.2	30.3	81.8	3.0
apes (78)	43.3	25.6	93.6	11.4	16.7	14.1	37.2	6.4
eyes (177)	82.6	80.7	58.0	65.3	32.2	28.3	49.2	18.1
girl (240)	43.3	30.0	0.0	37.5	5.4	1.3	0.0	1.3
great (219)	91.7	92.0	58.0	91.0	50.2	47.5	21.5	55.7
jungle (241)	49.3	53.7	0.0	37.6	27.4	24.1	0.0	18.3
tarzan (48)	41.3	66.7	30.9	7.7	39.6	25.0	60.4	47.9
time (145)	70.4	82.2	70.6	31.1	47.6	25.5	53.1	34.5
two (148)	41.0	92.3	84.6	91.7	10.8	8.1	7.4	14.9
way (101)	59.6	80.8	61.3	61.3	27.7	20.8	18.8	18.8
Micro-averaged	**53.3**	**55.4**	**28.2**	**34.3**	**27.9**	**22.2**	**22.8**	**22.5**

Chapter 12

Discussion and Future Work

In the second part of this thesis, we presented a new framework for data analysis. This framework generalizes the original single-sided IB principle. It enables one to define and solve a rich and novel family of optimization problems, which are all motivated by a single information theoretic principle, the multivariate IB principle. On the practical level, it suggests different ways to extract structure from data under a well defined theoretical framework.

We presented examples for several IB like constructions, including parallel IB, symmetric IB, and triplet IB. It should be clear, though, that future research could elucidate further problems and their applications.

Similarly to the single-sided IB-functional, the multivariate IB-functional is not convex with all of its arguments simultaneously. Thus, to construct solutions in practice one must employ different heuristics. We showed how to extend all the four algorithmic approaches suggested to the original IB principle into the multivariate scenario. We further demonstrated their usability to analyze real world data through different multivariate IB constructions. For each of these approaches, the specification of the algorithm is completed, once a specification of G_{in} and G_{out} is provided.

Much of the discussion related to the single-sided IB principle (Chapter 6) is relevant to the multivariate principle as well. For example, finite sample effects that were discussed in Section 6.1 might be even more acute when dealing with joint distributions over more than two random variables. Nonetheless, the alternative interpretation of the aIB and the sIB algorithms (relating them to the two-sample problem) is relevant for their multivariate extensions as well. The discussion regarding model selection issues and how to avoid over-fit (Section 6.2.2) is also naturally extended in our context. In particular, generalization considerations, similar to those suggested in [60] can be employed for estimating the maximal value of β (or the maximal number of clusters) that should be used.

Many possible connections with other data analysis methods merit further investigation. For example, the general structure of the multivariate iIB algorithm (Figure 10.1) is reminiscent of EM [24]. Moreover, as discussed in Appendix A there are strong relationships between the original IB problem and Maximum Likelihood estimation for mixture models. Hence, it is natural to look for further relationships between generative models and different multivariate IB problems. Specifically, this might suggest new generative models that are worth exploring. Other connections are, for example, to other dimensionality reduction techniques, such as Independent Component Analysis (ICA) [7]. The parallel IB provides an ICA-like decomposition with an important distinction. In contrast to ICA, it is aimed at preserving information about specific aspects of the data, defined by the user in specifying G_{out}.

The suggested multivariate IB framework addresses a rich family of optimization problems that involve minimization versus maximization of mutual information terms. However, this family is not complete in the sense that possible related problems are not captured by our formulation. The *discriminative IB* [19], which we already mentioned in Chapter 6 is one example. Recall that in this case the relevant information term (which we would like to maximize) is composed as a *difference* between two mutual information terms.

Nonetheless, it seems that a simple extension of our framework, where we define *three* networks: G_{in}, G^+_{out} and G^-_{out}, would be able to capture such discriminative problems as well. Specifically, in this case instead of minimizing $\mathcal{L}^{(1)} = \mathcal{I}^{G_{in}} - \beta \mathcal{I}^{G_{out}}$ we may consider the minimization of $\mathcal{I}^{G_{in}} - \beta^+ \mathcal{I}^{G^+_{out}} + \beta^- \mathcal{I}^{G^-_{out}}$, where β^+ and β^- are positive Lagrange multipliers, $\mathcal{I}^{G^+_{out}}$ refers to the information terms that we would like to maximize, and $\mathcal{I}^{G^-_{out}}$ refers to the ("irrelevant") information terms that we wish to minimize. Extending our theoretical analysis to handle this situation seems to be straightforward.

12.1 Future Research

There are many possible directions for future research. Below we mention several examples.

12.1.1 Multivariate relevance-compression function and specifying G_{in} and G_{out}

In the original IB problem the trade-off in the IB-functional is quantified by a single function, the relevance-compression function (Definition 2.3.1). As explained in Section 2.3 and illustrated in Figure 2.6, this function characterizes how well one can compress the variable X while preserving the information about the relevant variable Y. An important issue is to extend this discussion to the multivariate case. A possible way to do this is through the following definition.

Definition 12.1.1: The *multivariate relevance-compression function* for a given joint distribution $p(\mathbf{x})$ and a given specification of G_{in} and G_{out}, is defined as

$$\hat{\mathbf{R}}(\hat{\mathbf{D}}) \equiv \min_{\{\{\mathbf{p}(\mathbf{t_j}|\mathbf{u_j})\}_{\mathbf{j=1}}^{\mathbf{k}} : \mathcal{I}^{\mathbf{G}_{out}} \geq \hat{\mathbf{D}}\}} \mathcal{I}^{\mathbf{G}_{in}} , \tag{12.1}$$

where $p(\mathbf{x}, \mathbf{t}) \models G_{in}$ and the minimization is over all the normalized conditional distributions, $\{p(t_j \mid \mathbf{u}_j)\}_{j=1}^{k}$ for which the constraint is satisfied.

As in the single-sided IB case, this function separates between an achievable and a non-achievable region in a multivariate relevance-compression plane. In particular it is easy to verify that Definition 2.3.1 is a special case of this definition with G_{in} and $G^{(a)}_{out}$ of Figure 8.1. Additionally, is seems straightforward to extend Proposition 2.3.2, to show that, in general, $\hat{\mathbf{R}}(\hat{\mathbf{D}})$ is a non-decreasing concave function of $\hat{\mathbf{D}}$, where its slope determined through $\frac{\delta \hat{\mathbf{D}}}{\delta \hat{\mathbf{R}}} = \beta^{-1}$.

However, an important distinction is that this definition requires the specification of G_{in} and G_{out}. That is, given some joint distribution $p(\mathbf{x})$, there are many different possible (multivariate) relevance-compression functions, each one of them characterizes the "structure" in $p(\mathbf{x})$ in a different way. The underlying assumption in our formulation is that G_{in} and G_{out} are provided as part of the problem setup. Nonetheless, specifying these two networks might be far from trivial. For example, in the parallel IB case, where $\mathbf{T} = \{T_1, \ldots, T_k\}$, setting the "correct" value of k can be seen as a model selection task, and certainly not an easy one.

An important goal is to develop automatic methods for choosing "good" G_{in} and G_{out} specifications. Possible guidance can come from the above mentioned multivariate relevance-compression function. Specifically, it seems reasonable to prefer specifications that yield "better" relevance-compression curves, where "better" in our context means a higher curve in the multivariate relevance-compression plane (see Figure 2.6, right panel). Clearly, this issue calls for further research.

12.1.2 Parametric IB

Possible choices of G_{in} and G_{out} imply that our $\mathbf{T}$ variables will produce redundant (as opposed to compressed) representations of the observed $\mathbf{X}$ variables. Consider, for example, the parallel IB (Figure 8.2, upper panel) where $|\mathcal{T}| = \Pi_{j=1}^{k}|\mathcal{T}_j| \geq |\mathcal{X}|$. This is a typical situation for large enough k, even if every T_j has only two possible values (or clusters). In this construction, a trivial solution is available, where we assign each X value with some unique $\mathbf{T}$ value, and by that preserve all the relevant information about Y. Using a *parametric* variant of our framework serves to avoid these situations and make these cases challenging as well.

To achieve this we need to consider the alternative multivariate IB principle, $\mathcal{L}^{(2)}$, discussed in Section 8.2. Recall that in this formulation we aim to minimize $\mathcal{I}^{G_{in}}$ while at the same time minimize the KL divergence with respect to the target class, defined as the family of distributions which are consistent with G_{out}. [1] In principle, we might define this family not only through the independences implied by G_{out}, but also through some specific parametric form, which is further induced over $p(\mathbf{x}, \mathbf{t})$. In this case, minimizing $D_{KL}[p\|G_{out}]$ becomes a question of finding $p(\mathbf{x}, \mathbf{t})$ with minimum violation of the conditional independences implied by G_{out} and with the appropriate parametric form. In particular, this means that the number of free parameters can be drastically reduced, hence avoiding possible redundant solutions. A detailed discussion of this issue will be given elsewhere.

12.1.3 Relation to network information theory

The single-sided IB is intimately related to rate distortion theory, as we discussed in detail in Chapter 2. In particular, we noted in Section 6.2.1 that it might be possible to formulate the original IB problem through a "relevant-coding theorem", somewhat similarly to the rate distortion theorem.

Extending this discussion in our context, it seems that the multivariate IB principle is related to network information theory (see, e.g., [20], Chapter 14). This theory is concerned with the analysis of a communication system between many senders and receivers, that includes elements as cooperation, interference and feedback. The general problem of this theory can be stated as follows. Given a channel transition matrix which describes the effects of the interference and the noise in the network, decide whether or not the sources can be transmitted over the channel. This problem involves data compression as well as finding the capacity region of the network, and except for various special cases it has not yet been solved ([20], page 374).

Hence, the search for a "multivariate relevant-coding theorem", needs to be done on a shakier ground. While for the single-sided IB principle, the known rate distortion theorem can provide considerable guidance, this is not true for the multivariate IB. Nonetheless, an intriguing open question is whether it is possible to formulate the multivariate IB principle through a "multivariate relevant-coding" theorem. Obviously such a formulation will require a definition of a "multivariate relevant code", associated with a multivariate relevant-distortion term which can be derived directly from $p(\mathbf{x}, \mathbf{t})$ (and will be related to $\mathcal{I}^{G_{out}}$ in our context). As in the single-sided IB case, this issue obviously requires a separate investigation, and is beyond the scope of this thesis. Nonetheless, we note here that such a rigorous formulation of the multivariate IB principle might provide some hints in regard to open problems in network information theory.

[1] Note that this KL minimization is in general different from the standard KL minimization in maximum likelihood estimation. See Section A.5 for a discussion.

Epilogue

The volume of available data in a variety of domains has grown rapidly over the last few years. Examples include the consistent growth in the amount of on-line text due to the expansion of the World Wide Web, and the dramatic increase in the available genomic information due to the development of new technologies for gathering such data. As a result, there is a crucial need for complex data analysis methods. A major goal in this context is the development of new unsupervised dimensionality reduction methods that serve to reveal the inherent hidden structure in a given body of complex data. One important class of such methods are clustering techniques. Although numerous clustering algorithms exist, typically, the results they generate are hard to interpret. Clearly, a sound interpretation should arise from a combination of a clear intuition on the one hand, accompanied by well defined theoretical groundwork on the other. We argue that the IB method, as described in this thesis, responds satisfactorily to both criteria.

More specifically, the IB principle leads to a rich theoretical framework which is nicely analogous, and in some sense unifies, the well established rate distortion theory on one hand, as well as some aspects of channel coding theory on the other. At the same time, the basic idea is simple and intuitive: We seek clusters that are as informative (as possible) about some predefined target, or relevant variable. In particular, we argue that this prerequisite of specifying the relevant variable in advance, suggests a natural scheme of posing clustering problems which immediately leads to an objective interpretation of the resulting clusters in terms of the information they capture about this relevant variable.

The first half of this thesis makes several contributions. First, we provide a primary detailed review of the original IB method. Second, we suggest new algorithms that prove to be crucial in order to construct solutions in practice to the IB problem. Last, we describe rich empirical evidence that establish the method as a major data analysis approach that successfully competes, and usually outperforms previous standard methods.

The contributions of the second half of this thesis are as follows. First, we fully extend the theory of the original IB framework to cope with any finite number of variables. We further extend all the algorithmic approaches suggested for the original problem to handle multivariate IB constructions. Finally, we demonstrate the applicability of these multivariate algorithms in solving different data analysis tasks over a variety of real world datasets.

This multivariate formulation defines a rich family of novel optimization problems which are all unified under a single information-theoretic principle, the multivariate IB principle. In particular this allows us to extract structure from data in many different ways. In the second part of this thesis we investigated only three examples, but we believe that this is only the tip of the iceberg.

An immediate corollary of this analysis is that the general term of *clustering* conceals a broad family of many distinct problems which deserve special consideration. To the best of our knowledge, the multivariate IB framework described in this thesis is the first successful attempt to define these sub-problems, solve them, and demonstrate their importance.

Part IV

Bibliography

Bibliography

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Nat. Acad. Sci. USA*, 96:6745-6750, 1999.

[2] R. Apweiler et al. InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucl. Acids Res.*, 29(1):37-40, 2001.

[3] S. Arimoto. An algorithm for calculating the capacity of an arbitrary discrete memoryless channel. *IEEE Trans. Inform. Theory*, IT-18:14-20, 1972.

[4] T. K. Attwood et al. PRINTS and PRINTS-S shed light on protein ancestry. *Nucleic Acids Res.*, Vol. 30, No. 1, 2002.

[5] R. Baierlein. *Atoms and information theory*. W. H. Freeman and company, San Francisco, 1971.

[6] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. *Proc. of the 21th Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1998.

[7] A. J. Bell and T .J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neur. Comp.* 7, 1129–1159, 1995.

[8] S. Belongie, J. Malik, and J. Puzicha . Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Machine Intell.*, 24(4): 509-522, 2002.

[9] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *Journal of Computational Biology*, 7:559–584, 2000.

[10] W. Bialek, I. Nemenman, and N. Tishby. Predictability, complexity and learning. *Neural Computation* 13, 2409–2462, 2001.

[11] W. Bialek and N. Tishby. Extracting relevant information. Unpublished manuscript, 1999.

[12] Y. Bilu, M. Linial, N. Slonim, and N. Tishby. Locating transcription factors binding sites using a variable memory Markov model. Unpublished manuscript, 2001.

[13] C. M. Bishop. *Neural Networks for pattern recognition*. Clarenodon pres - Oxford, 1995.

[14] R. E. Blahut. Computation of channel capacity and rate distortion function. *IEEE Trans. Inform. Theory*, IT-18:460-473, 1972.

[15] R. E. Blahut. Hypothesis testing and information theory. *IEEE Trans. Inform. Theory*, IT-20:405-417, 1974.

[16] R. E. Blahut. *Principles and practice of information theory.* Addison-Wesley Publishing, 1987.

[17] M. Blatt, M. Wiesman, and E. Domany. Data clustering using a model granular magnet. *Neural Computation* 9, 1805-1842, 1997.

[18] P. S. Bradley and U. M. Fayyad. Refining initial points for K-Means clustering. *Proc. 15th International Conf. on Machine Learning*, 1998.

[19] G. Chechik and N. Tishby. Extracting relevant information with side information. *Advances in Neural Information Processing Systems (NIPS) 15*, 2002, to appear.

[20] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley and Sons, New York, 1991.

[21] I. Csiszár. On the computation of rate distortion functions. *IEEE Trans. Inform. Theory*, IT-20:122-124, 1974.

[22] I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *Annals of Probability*, 3(1):146–158, 1975.

[23] I. Csiszár and G. Tusnády. Information geometry and alternating minimization procedures. *Statistics and Decisions*, Supplement Issue 1:205-237, 1984.

[24] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, Vol. 39, pp. 1-38, 1977.

[25] I. S. Dhillon, S. Mallela, and R. kumar. A divisive information theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research (JMLR)*, Special issue on variable and feature selection, to appear.

[26] B. E. Dom. An information-theoretic external cluster-validity measure. *IBM Research Report RJ 10219*, 10/5/2001.

[27] K. Eguchi. Adaptive cluster-based browsing using incrementally expanded queries and its effects. *Proc. of the 22rd Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1999.

[28] M. Eisen, P. Spellman, P. Brown and D. Botstein. Cluster analysis and display of genome wide expression patterns. *Proc. Nat. Acad. Sci. USA*, 95, 14863–14868, 1998.

[29] R. El-Yaniv, S. Fine, and N. Tishby. Agnostic classification of Markovian sequences. *Advances in Neural Information Processing Systems (NIPS) 10*, pp. 465–471, 1997.

[30] R. El-Yaniv and O. Souroujon. Iterative double clustering for unsupervised and semi-supervised learning. *Advances in Neural Information Processing Systems (NIPS) 14*, 2001.

[31] R. A. Fisher. The use of multiple measurements in taxonomic problems *Annual Eugenics*, 7, Part II, 179-188, 1936.

[32] N. Friedman, O. Mosenzon, N. Slonim, and N. Tishby. Multivariate information bottleneck. *Proc. of the 17th conf. on Uncertainty in artificial Intelligence (UAI-17)*, 2001.

[33] M. R. Garey, D. S. Johnson, and H. S. Witsenhausen. The complexity of the generalized Lloyd-Max problem. *IEEE Trans. Inform. Theory*, 28(2):255–256, 1982.

[34] Y. Gdalyahu, D. Weinshall, and M. Werman, Randomized algorithm for pairwise clustering. *Advances in Neural Information Processing Systems (NIPS) 11*, pp. 424–430, 1998.

[35] T. Gedeon, A. E. Parker, and A .G .Dimitrov. Information distortion and neural coding. Submitted to the Canadian Applied Math Quarterly.

[36] Z. Gilula and A. M. Krieger. Collapsed two-way contingency table and the Chi-square reduction principle. *Journal of the Royal Statistical Society*, Series B, 51(3):425–433, 1989.

[37] A. Globerson and N. Tishby. Sufficient dimensionality reduction. *Journal of Machine Learning Research (JMLR)*, Special issue on variable and feature selection, to appear.

[38] J. Goldberger, H. Greenspan, and S. Gordon. Unsupervised image clustering using the information bottleneck method. *The annual symposium for Pattern Recognition of the DAGM02*, Zurich, 2002.

[39] T. Golub, D. Slonim, P. Tamayo, C.M. Huard, J.M. Caasenbeek, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, Vol. 286, 531–537, 1999.

[40] A. Gordon. *Classification* (2nd edition). Chapman and Hall/CRC press, London, 1999.

[41] M. Gorodetsky. *Methods for discovering semantic relations between words based on co-occurrence patterns in corpora.* Masters thesis, School of Computer Science and Engineering, Hebrew university, 2002.

[42] M. Gutman. Asymptotically optimal classification for multiple tests with empirically observed statistics. *IEEE Trans. Inform. Theory*, Vol. 35, No. 2, pp. 401–408, 1989.

[43] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: scatter/gather on retrieval results. *Proc. of the 19th Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1996.

[44] T. Hofmann, J. Puzicha, and M. I. Jordan. Learning from dyadic data. *Advances in Neural Information Processing Systems (NIPS) 11*, 1998.

[45] T. Hofmann. Probabilistic latent semantic indexing. *Proc. of the 22nd Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp. 50–57, 1999.

[46] E. T. Jayens. Information theory and statistical mechanics. *Physical Review*, 106, 620–630, 1957.

[47] K. Lang. Learning to filter netnews. *Proc. 12th International Conf. on Machine Learning*, pp. 331–339, 1995.

[48] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, Vol. 401, pp. 788–791, 1999.

[49] E. L. Lehmann. *Testing statistical hypotheses*. John Wiley and sons, New-York, 1959.

[50] J. Lin. Divergence measures based on the Shannon entropy. *IEEE Trans. Inform. Theory*, 37(1):145–151, 1991.

[51] D. J. C. MacKay. *Information theory, inference, and learning algorithms*. Available at *http://wol.ra.phy.cam.ac.uk/itprnn/book.ps.gz*, Draft 3.1.2, October 20, 2002.

[52] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. Available at *http://www.cs.cmu.edu/ mccallum/bow*, 1996.

[53] G. J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley and Sons, New York, 2000.

[54] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan (editor), *Learning in Graphical Models*, pp. 355–368, Dordrecht: Kluwer Academic Publishers, 1998.

[55] I. Nemenman and N. Tishby. Network information theory. In preparation.

[56] Y. Ofran and H. Margalit. Is there a relationship between the amino acid composition of a protein and its fold? Submitted.

[57] L. M. Optican, T. J. Gawne, B. J. Richmond, and P. J. Joseph. Unbiased measures of transmitted information and channel capacity from multivariate neuronal data. *Biol. Cybernet.*, 65, pp. 305–310, 1991.

[58] A. E. Parker, T. Gedeon, A. G. Dimitrov, and B. Roosien. Annealing and the rate distortion problem. *Advances in Neural Information Processing Systems (NIPS) 15*, 2002, to appear.

[59] J. Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kauffman, San Francisco, 1988.

[60] F.C. Pereira, N. Tishby, and L. Lee. Distributional clustering of English words. *In 30th Annual Meeting of the Association for Computational Linguistics*, pp. 183–190, 1993.

[61] J. Puzicha, T. Hofmann, and J. M. Buhmann. Histogram clustering for unsupervised segmentation and image retrieval. *Pattern Recognition Letters*:20(9), 899-909, 1999.

[62] J. Rissanen. Modeling by shortest data description. *Automatica*, 14, pp. 465–471, 1978.

[63] K. Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE*, 86:2210–2239, 1998.

[64] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, Vol. 290, no. 5500, pp. 2323–2326, 2000.

[65] G. Salton. Developments in automatic text retrieval. *Science*, Vol. 253, pp. 974–980, 1990.

[66] R. E. Schapire and Y. E. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39, 2000.

[67] E. Schneidman, W. Bialek, and M. J. Berry: An information theoretic approach to the functional classification of neurons. *Advances in Neural Information Processing Systems (NIPS) 15*, 2002, to appear.

[68] E. Schneidman, N. Slonim, R. R. de Ruyter van Steveninck, N. Tishby, and W. Bialek. Analyzing neural codes using the information bottleneck method. Unpublished manuscript, 2001.

[69] A. Schreibman, R. El-Yaniv, S. Fine and N. Tishby. On the two-sample problem and the Jensen-Shannon divergence for Markov sources. In preparation.

[70] C. E. Shannon. A mathematical theory of communication. *The Bell Systems Technical Journal*, Vol. 27, pp. 379–423, 623–656, 1948.

[71] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[72] N. Slonim, G. Bejerano, S. Fine, and N. Tishby. Discriminative feature selection via multiclass variable memory Markov model. *EURASIP Journal on Applied Signal Processing (JASP)*, Special issue on Unstructured Information Management from Multimedia Data Sources, to appear.

[73] N. Slonim, N. Friedman, and N. Tishby Agglomerative multivariate information bottleneck. *Advances in Neural Information Processing Systems (NIPS) 14*, 2001.

[74] N. Slonim, N. Friedman, and N. Tishby Unsupervised document classification using sequential information maximization. *Proc. of the 25th Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2002.

[75] N. Slonim, R. Somerville, N. Tishby, and O. Lahav. Objective classification of galaxies spectra using the information bottleneck method. *Monthly Notices of the Royal Astronomical Society (MNRAS)*, 323, 270, 2001.

[76] N. Slonim and N. Tishby. Agglomerative information bottleneck. *Advances in Neural Information Processing Systems (NIPS) 12*, pp. 617–623, 1999.

[77] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. *Proc. of the 23rd Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, pp. 208–215, 2000.

[78] N. Slonim and N. Tishby. The power of word clusters for text classification. *In 23rd European Colloquium on Information Retrieval Research*, 2001.

[79] N. Slonim and Y. Weiss. Maximum likelihood and the information bottleneck. *Advances in Neural Information Processing Systems (NIPS) 15*, 2002, to appear.

[80] M. Studen and J. Vejnarova The Multi-information function as a tool for measuring stochastic dependence. In M. I. Jordan (editor), *Learning in Graphical Models*, pp. 261–298, Dordrecht: Kluwer Academic Publishers, 1998.

[81] K. Takabatake. Linear separation theorem in distributional clustering. *IJCNN*, pp.88–92, 2001.

[82] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. *Proc. 37th Allerton Conference on Communication and Computation*, 1999.

[83] N. Tishby and N. Slonim. Data clustering by Markovian relaxation and the information bottleneck method. *Advances in Neural Information Processing Systems (NIPS) 13*, 2000.

[84] A. Treves and S. Panzeri. The upward bias in measures of information derived from limited data samples. *Neural Computation*, 7:399-407, 1995.

[85] C. J. van Rijsbergen. *Information retrieval*. London: Butterworths; 1979.

[86] V. N. Vapnik. *The nature of statistical learning theory*. Springer, 1999.

[87] J. J. Verbeek. *An information theoretic approach to finding word groups for text classification*. Masters thesis, The Institute for Logic, Language and Computation, University of Amsterdam, 2000.

[88] A. k. C. Wong and M. You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-7, no. 5, pp. 599–609, 1985.

[89] O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. *Proc. of the 21th Ann. Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 1998.

Part V

Appendices

Appendix A

Maximum Likelihood and the Information Bottleneck

The IB method provides an information theoretic formulation to address clustering problems. However, a standard and well established approach to clustering is Maximum likelihood (ML) of mixture models. In this appendix, following [79], we investigate how the two methods are related.

In mixture modeling we assume the measurements y for each x come from one of $|\mathcal{T}|$ possible statistical sources, each with its own parameters Θ_t (e.g. μ_t, σ_t in Gaussian mixtures). Clustering corresponds to first finding the maximum likelihood estimates of Θ_t and then using these parameters to calculate the posterior probability that the measurements at x were generated by each source. These posterior probabilities define a "soft" clustering of $\mathcal{X}$.

While the ML and the IB approaches try to solve the same problem, the viewpoints are quite different. In the information theoretic approach no assumption is made regarding how the data were generated but we assume that the joint distribution $p(x, y)$ is known exactly. In the maximum likelihood approach we assume a specific generative model for the data and assume we have samples $n(x, y)$, not the true probability.

In spite of these conceptual differences we show that under a proper choice of the generative model, these two problems are strongly related. Specifically we use the multinomial mixture model (a.k.a. the one-sided clustering model [44] or the asymmetric clustering model [61]), and provide a simple "mapping" between the concepts of one problem to the concepts of the other. Using this mapping we show that in general, searching for a solution to one problem induces a search in the solution space of the other. Furthermore, for uniform input distribution over X or for large sample sizes, we show that the problems are mathematically equivalent. Specifically, in these cases, every fixed point of the IB-functional defines a fixed point of the likelihood and vice versa. Moreover, the values of the functionals at the fixed points are equal under simple linear transformations. As a result, in these cases, every algorithm that solves one of the problems induces a solution to the other.

A.1 Short review of ML for mixture models

In the Gaussian mixture model we generate an observation y at index x by first choosing a label $t(x)$ by sampling from $\pi(t)$ and then sampling $y(x)$ from a Gaussian with mean $\mu_{t(x)}$ and variance $\sigma^2_{t(x)}$. In a multinomial mixture model, we assume that y takes on discrete values and sample it from a multinomial distribution $\theta(y|t(x))$. In the one-sided clustering model [44] [61] we further assume that there can be multiple observations y corresponding to a single x but they are all sampled from the same multinomial distribution. This model can be described through the following generative process:

- for each x choose a unique label $t(x)$ by sampling from $\pi(t)$.

- For $k = 1 : N$
  - choose x_k by sampling from $\gamma(x)$.
  - choose y_k by sampling from $\theta(y|t(x_k))$ and increase $n(x_k, y_k)$ by one.

Note that in this model, observations (x_k, y_k) for a specific $x \in \mathcal{X}$ are conditionally independent given $t(x)$. Let $\vec{t} = (t, ..., t_{|\mathcal{X}|})$ denote the random vector that defines the (typically hidden) labels for all $x \in \mathcal{X}$. The complete likelihood is given by:

$$\begin{aligned} p(x, y, \vec{t} : \pi, \theta, \gamma) &= \Pi_{i=1}^{|\mathcal{X}|}\pi(t(x_i))\Pi_{k=1}^{N}\gamma(x_k)\theta(y_k|t(x_k)) && \text{(A.1)} \\ &= \Pi_{i=1}^{|\mathcal{X}|}\pi(t(x_i))\Pi_{i=1}^{|\mathcal{X}|}\Pi_{j=1}^{|\mathcal{Y}|}[\gamma(x_i)\theta(y_j|t(x_i))]^{n(x_i,y_j)} \ , && \text{(A.2)} \end{aligned}$$

where $n(x_i, y_j)$ is a count matrix.

The (true) likelihood is defined through summing over all the possible choices of $\vec{t}$,

$$L(n(x, y); \pi, \theta, \gamma) = \sum_{\vec{t}} p(x, y, \vec{t} : \pi, \theta, \gamma) \ . \qquad \text{(A.3)}$$

Given $n(x, y)$, the goal of ML estimation is to find an assignment for the parameters $\pi(t), \theta(y \mid t)$ and $\gamma(x)$ such that this likelihood is (at least locally) maximized. Since it is easy to show that the ML estimate for $\gamma(x)$ is just the empirical counts $n(x)/N$, we further focus only on estimating π, θ.

A standard algorithm for this purpose is the EM algorithm [24]. Informally, in the E-step we replace the missing value of $t(x)$ by its distribution $p(t(x)|y(x))$ which we denote here by $p_x(t)$. In the M-step we use that distribution to reestimate π, θ. Using standard derivation it is easy to verify that in our context the E-step is defined through

$$\begin{aligned} p_x(t) &= k(x)\pi(t)e^{\sum_y n(x,y)\log\theta(y|t)} && \text{(A.4)} \\ &= k(x)\pi(t)e^{n(x)\sum_y n(y|x)\log\theta(y|t)} && \text{(A.5)} \\ &= k_2(x)\pi(t)e^{n(x)[\sum_y n(y|x)\log\theta(y|t) - \sum_y n(y|x)\log n(y|x)]} && \text{(A.6)} \\ &= k_2(x)\pi(t)e^{-n(x)D_{KL}[n(y|x)\|\theta(y|t)]} \ , && \text{(A.7)} \end{aligned}$$

where $k(x)$ and $k_2(x)$ are normalization factors. The M-step is simply given by

$$\begin{cases} \pi(t) \propto \sum_x p_x(t) \\ \theta(y \mid t) \propto \sum_x n(x, y)p_x(t) \ . \end{cases} \qquad \text{(A.8)}$$

Iterating over these EM steps is guaranteed to converge to a local fixed point of the likelihood. Moreover, every fixed point of the likelihood defines a fixed point of this algorithm.

An alternative derivation [54] is to define the free energy functional:

$$\begin{aligned} F(n(x, y) : q, \pi, \theta) &= -\sum_{t,x} p_x(t)\left[\log\pi(t) + \sum_y n(x, y)\log\theta(y \mid t)\right] && \text{(A.9)} \\ &\quad + \sum_{t,x} p_x(t)\log p_x(t) \ . && \text{(A.10)} \end{aligned}$$

The E-step then involves minimizing F with respect to q while the M-step minimizes it with respect to π, θ. Since this functional is bounded (under mild conditions), the EM algorithm will converge to a local fixed point of the free energy which corresponds to a fixed point of the likelihood. At these fixed points, the free energy will become identical to $-\log L(n(x, y) : \pi, \theta)$.

A.2 The ML $\leftrightarrow$ IB mapping

As already mentioned, the IB problem and the ML problem stem from different motivations and involve different settings. Therefore, it is not entirely clear what is the purpose of mapping between these problems. For our needs this mapping is defined to achieve two goals. The first is theoretically motivated: using the mapping we will show some mathematical equivalence between both problems. The second is practically motivated, where we will show that algorithms designed for one problem are (in some cases) suitable for solving the other.

A natural mapping would be to identify each distribution with its corresponding one. However, this direct mapping is problematic. Assume that we are mapping from ML to IB. If we directly map $p_x(t), \pi(t), \theta(y \mid t)$ to $p(t \mid x), p(t), p(y \mid t)$, respectively, obviously there is no guarantee that the IB Markovian relation (Eq. (2.11)) will hold once we complete the mapping. Specifically, using this assumption to extract $p(t)$ through Eq. (2.12) will in general result in a different prior over T, then by simply defining $p(t) = \pi(t)$. However, once we define $p(t \mid x)$ and $p(x, y)$, the other distributions can be extracted through the "IB-step" defined in Eqs. (2.12). Moreover, as already shown in Section 3.1.1 performing this step can only improve (decrease) the corresponding IB-functional.

A similar phenomenon is present once we map from IB to ML. Although in principle there are no "consistency" problems by mapping directly, we know that once we define $p_x(t)$ and $n(x, y)$, we can extract π and θ by a simple M-step. This step, by definition, will only improve the likelihood, which is our goal in this setting.

The only remaining issue is to define a corresponding component in the ML setting to the trade-off parameter β of the IB problem. As we will show in the next section, the natural choice for this purpose is the sample size, $N = \sum_{x,y} n(x, y)$.

Therefore, to summarize, we define the $ML \leftrightarrow IB$ mapping by

$$p_x(t) \leftrightarrow p(t \mid x), \;\; \frac{1}{N} n(x, y) \leftrightarrow p(x, y), \;\; N \leftrightarrow r\beta \;, \tag{A.11}$$

where r is a (scaling) constant and the mapping is completed by performing an IB-step or an M-step according to the mapping direction. Given this mapping, every search in the solution space of the IB problem induces a search in the solution space of the ML problem, and vice versa.

Observation A.2.1 : *When X is uniformly distributed (i.e., $n(x)$ or $p(x)$ are constant), the $ML \leftrightarrow IB$ mapping is equivalent for a direct mapping of each distribution to its corresponding one.*

This observation stems directly from the fact that if X is uniformly distributed, then the IB-step defined in Eqs. (2.12) and the M-step defined in Eqs. (A.8) are mathematically equivalent.

Observation A.2.2 : *When X is uniformly distributed, the EM algorithm is equivalent to the iterative IB (iIB) algorithm under the $ML \leftrightarrow IB$ mapping with* $r = |X|$.

Again, this observation is a direct result of the equivalence of the IB-step and the M-step for uniform prior over X. Additionally, in this case $n(x) = \frac{N}{|\mathcal{X}|} = \frac{N}{r} = \beta$, hence Eq. (A.7) and Eq. (2.16) are also equivalent.

It is important to emphasize, though, that this equivalence only holds for a specific choice of $\beta = n(x)$. While clearly the iIB algorithm (and the IB problem in general) are meaningful for any value of β, there is no such freedom (for good or worse) in the ML setting, and the exponential factor in EM *must* be $n(x)$.

A.3 Comparing ML and IB

A.3.1 Comparison for uniform $p(x)$

Theorem A.3.1 : *When X is uniformly distributed and $r = |\mathcal{X}|$, all the fixed points of the likelihood L are mapped to all the fixed points of the IB-functional $\mathcal{L}$ with $\beta = n(x)$. Moreover, at the fixed points, $-\log L \propto \mathcal{L} + c$, with c constant.*

Corollary A.3.2: *When X is uniformly distributed, every algorithm which finds a fixed point of L, induces a fixed point of $\mathcal{L}$ with $\beta = n(x)$, and vice versa. When the algorithm finds several different fixed points, the solution that maximizes L is mapped to the solution that minimizes $\mathcal{L}$.*

Proof: We prove the direction from ML to IB. The opposite direction is similar. We assume that we are given observations $n(x, y)$ where $n(x)$ is constant, and π, θ that define a fixed point of the likelihood L. As a result, this is also a fixed point of the EM algorithm (where $p_x(t)$ is defined through an E-step). Using Observation A.2.2 it follows that this fixed-point is mapped to a fixed-point of $\mathcal{L}$ with $\beta = n(x)$, as required.

Since at the fixed point, $-\log L = F$, it is enough to show the relationship between F and $\mathcal{L}$. Rewriting F from Eq. (A.9) we get

$$F(n(x,y) : q, \pi, \theta) = \sum_{t,x} p_x(t) \log \frac{p_x(t)}{\pi(t)} - \sum_{t,y} \log \theta(y \mid t) \sum_x n(x,y) p_x(t) \; . \tag{A.12}$$

Using the $ML \rightarrow IB$ mapping and Observation A.2.1 we get

$$F = \sum_{t,x} p(t \mid x) \log \frac{p(t \mid x)}{p(t)} - r\beta \sum_{t,y} \log p(y \mid t) \sum_x p(x,y) p(t \mid x) \; . \tag{A.13}$$

Multiplying both sides by $p(x) = \frac{1}{|\mathcal{X}|} = r^{-1}$ and using the IB Markovian independence relation, we find that

$$r^{-1}F \;\; = \;\; \sum_{t,x} p(x) p(t \mid x) \log \frac{p(t \mid x)}{p(t)} - \beta \sum_{t,y} p(t) p(y \mid t) \log p(y \mid t) \; . \tag{A.14}$$

Reducing a (constant) $\beta H(Y) = -\beta \sum_{t,y} p(t) p(y \mid t) \log p(y)$ to both sides gives:

$$r^{-1}F - \beta H(Y) = I(T;X) - \beta I(T;Y) = \mathcal{L} \; , \tag{A.15}$$

as required. We emphasize again that this equivalence is for a specific value of $\beta = n(x)$. ▮

Corollary A.3.3: *When X is uniformly distributed and $r = |\mathcal{X}|$, every algorithm decreases F, if and only if it decreases $\mathcal{L}$ with $\beta = n(x)$.*

This corollary is a direct result of the above proof that showed the equivalence of the free energy of the model and the IB functional (up to linear transformations).

A.3.2 Comparison for large sample size

The previous section dealt with the special case of uniform prior over X. In the following we provide similar results for the general case, when N (or β) are large enough.

Theorem A.3.4: *For $N \to \infty$ (or $\beta \to \infty$), all the fixed points of L are mapped to all the fixed points of $\mathcal{L}$, and vice versa. Moreover, at the fixed points, $-\log L \propto \mathcal{L} + c$, with c constant.*

Corollary A.3.5: *When $N \to \infty$ every algorithm which finds a fixed point of L induces a fixed point of $\mathcal{L}$ with $\beta \to \infty$, and vice versa. When the algorithm finds several different fixed points, the solution that maximizes L is mapped to the solution that minimize $\mathcal{L}$.*

Proof: Again, we only prove the direction from ML to IB as the opposite direction is similar. We are given $n(x, y)$ where $N = \sum_{x,y} n(x, y) \to \infty$ and π, θ that define a fixed point of L. Using the E-step in Eq.(A.7) we extract $p_x(t)$, ending up with a fixed point of the EM algorithm. From $N \to \infty$ follows $n(x) \to \infty \ \ \forall x \in \mathcal{X}$. Therefore, the mapping $p_x(t)$ becomes deterministic:

$$p_x(t) = \begin{cases} 1 & t = argmin_{t'} D_{KL}[n(y|x) \| \theta(y|t')] \\ 0 & \text{otherwise.} \end{cases} \tag{A.16}$$

Performing the $ML \to IB$ mapping (including the IB-step), it is easy to verify that we get $p(y \mid t) = \theta(y \mid t)$ (but $p(t) \neq \pi(t)$ if the prior over X is not uniform). After completing the mapping we try to update $p(t \mid x)$ through Eq.(2.16). Since now $\beta \to \infty$ it follows that $p(t \mid x)$ will remain deterministic. Specifically,

$$p^{new}(t \mid x) = \begin{cases} 1 & t = argmin_{t'} D_{KL}[p(y \mid x) \| p(y|t')] \\ 0 & \text{otherwise,} \end{cases} \tag{A.17}$$

which is equal to its previous value. Therefore, we are at a fixed point of the iIB algorithm, and by that at a fixed point of the IB functional $\mathcal{L}$, as required.

To show that $-\log L \propto \mathcal{L} + c$ we note again that at the fixed point $F = -\log L$. From Eq.(A.12) we see that

$$\lim_{N \to \infty} F = -\sum_{t,y} \log \theta(y \mid t) \sum_x n(x, y) p_x(t) \ . \tag{A.18}$$

Using the $ML \to IB$ mapping and similar algebra as above, we find that

$$\lim_{N \to \infty} F = -r\beta I(T; Y) + r\beta H(Y) = \lim_{\beta \to \infty} r(\mathcal{L} + \beta H(Y)) \ . \tag{A.19}$$

∎

Corollary A.3.6: *When $N \to \infty$ every algorithm decreases F, if and only if it decreases $\mathcal{L}$ with $\beta \to \infty$.*

How large must N (or β) be? We address this question through numeric simulations in the next section. However, roughly speaking, the value of N for which the above results (approximately) hold is related to the "amount of uniformity" in $n(x)$. Specifically, a crucial step in the above proof assumed that each $n(x)$ is large enough such that $p_x(t)$ becomes deterministic. Clearly, when $n(x)$ is less uniform, achieving this situation requires larger N values.

A.4 Simulations

We performed several different simulations using different IB and ML algorithms. Due to lack of space, only one example is reported below. In this example we used the *Multi*10_1 subset of the $20NG$ corpus [47], consisting of 500 documents randomly chosen from ten different discussion groups (see Section 4.5.1). Denoting the documents by X and the words by Y, after the pre-processing described in Section 4.5.1 we have $|\mathcal{X}| = 500, \ |\mathcal{Y}| = 2000, \ N = 43{,}433, \ |\mathcal{T}| = 10$.

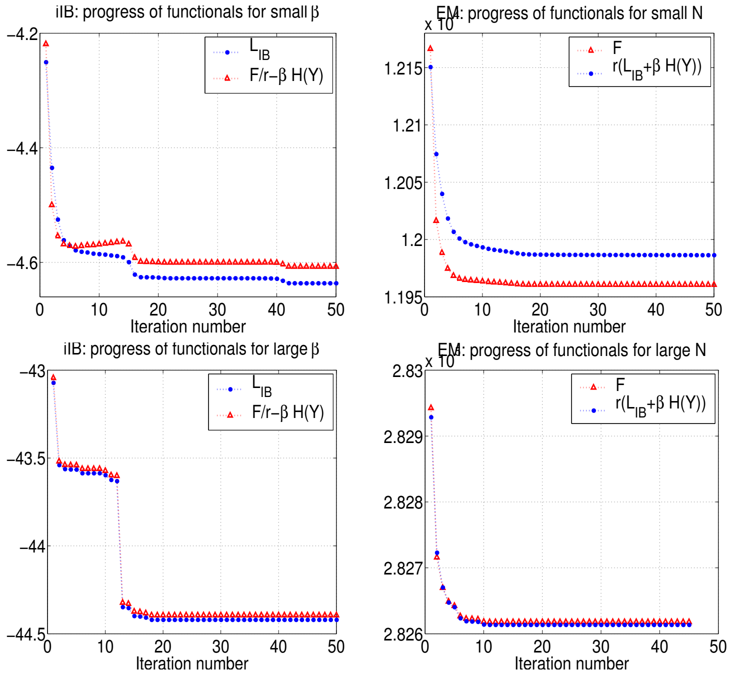

Figure A.1: Progress of $\mathcal{L}$ and F for different β and N values, while running iIB and EM.

Since our main goal was to test for differences between IB and ML for different values of N (or β), we further produced another dataset. In these data we randomly choose only about 5% of the word occurrences for every document $x \in \mathcal{X}$, ending up with $N = 2,171$.

For both datasets we clustered the documents into ten clusters, using both EM and the iIB algorithm (where we took $p(x,y) = \frac{1}{N}n(x,y)$, $\beta = \frac{N}{r}$, $r = |\mathcal{X}|$). For each algorithm we used the $ML \leftrightarrow IB$ mapping to calculate F and $\mathcal{L}$ during the process (e.g., for iIB, after each iteration we mapped from IB to ML, including the M-step, and calculated F). We repeated this procedure for 100 different initializations, for each dataset.

In these 200 restarts, we found that usually both algorithms improved both functionals monotonically. Comparing the functionals during the process, we see that for the smaller sample size the differences are indeed more evident (Figure A.1). Comparing the final values of the functionals (after 50 iterations, which typically was enough for convergence), we see that in 58 out of 200 runs, iIB converged to a smaller value of F than EM, and in 46 runs, EM converged to a smaller value of $\mathcal{L}$. Hence, in some cases, iIB finds a better ML solution or EM finds a better IB solution. We note that this phenomenon was much more common for the large sample size case.

A.5 Discussion

While we have shown that the ML and IB approaches are equivalent under certain conditions, it is important to keep in mind the different assumptions both approaches make regarding the joint distribution over x, y, t. The mixture model (1) assumes that Y is independent of X given $T(X)$ and (2) assumes that $p(y \mid x)$ is one of a small number ($|\mathcal{T}|$) of possible conditional distributions. For this reason, the marginal probability over x, y (i.e., $p(x,y) : \pi, \theta)$) is usually different from $\hat{p}(x,y) = \frac{1}{N}n(x,y)$. Indeed, an alternative view of ML estimation is as minimizing $D_{KL}[\hat{p}(x,y) \| L(n(x,y) : \pi, \theta)]$.

On the other hand, in the IB framework, T is *defined* through the IB Markovian independence relation: $T \leftrightarrow X \leftrightarrow Y$. Therefore, the solution space is the family of distributions for which this relation holds and the marginal distribution over x, y is consistent with the input. Interestingly, it is possible to give an alternative formulation for the IB problem which also involves KL minimization (see Section 8.2). In this formulation the IB problem is related to minimizing $D_{KL}[p(x,y,t)\|Q(x,y,t)]$, where $Q(x,y,t)$ denotes the family of distributions for which the *mixture model* assumption holds, $X \leftrightarrow T \leftrightarrow Y$. [1]

In this sense, we may say that while solving the IB problem, we are trying to minimize the KL with respect to the "ideal" world, in which T separates X from Y (and, thus, preserving all the information about Y). On the other hand, while solving the ML problem, we assume an "ideal" world, and try to minimize the KL with respect to the given marginal distribution $\hat{p}(x,y)$. Our theoretical analysis shows that under the $ML \leftrightarrow IB$ mapping, these two procedures are in some cases equivalent.

Once we are able to map between ML and IB, it should be interesting to try to adopt additional concepts from one approach to the other. In the following we provide two such examples. In the IB framework, for large enough β, the quality of a given solution is measured through $\frac{I(T;Y)}{I(X;Y)} \leq 1$. This measure provides a theoretical upper bound, which can be used for purposes of model selection and more. Using the $ML \leftrightarrow IB$ mapping, we can now adopt this measure for the ML estimation problem (for large enough N); In EM, the exponential factor $n(x)$ in general depends on x. However, its analogous component in the IB framework, β, obviously does not. Nonetheless, in principle it is possible to reformulate the IB problem while defining $\beta = \beta(x)$ (without changing the form of the optimal solution). We leave this issue for future research.

We have shown that for the multinomial mixture model, ML and IB are equivalent in some cases. It is worth noting that in principle, by choosing a different generative model, one may find further equivalences. Additionally, in Part III we described (and solved) a new family of IB-like variational problems. A natural question is to look for further generative models that can be mapped to these multivariate IB problems, and we are working in this direction.

[1] Recall that the KL with respect to the *family* Q is defined as the minimum over all the members in Q. Therefore, here, both arguments of the KL change during the process, and the distributions involved in the minimization are over all the three random variables.

Appendix B

Cluster accuracy and mutual information

In Section 4.5 we provided empirical evidence suggesting that IB algorithms (and in particular sIB) can extract highly "accurate" document clusters, in an entirely unsupervised manner. More precisely, the clusters extracted by sIB were typically well correlated with the existing topics in the corpus.

Nonetheless, these results call for further investigation. On the one hand, the sIB algorithm tries to maximize the mutual information between the document clusters and the *words* (i.e., the features) appearing in these documents, $I(T_d; W)$. On the other hand, the extracted clusters are found to be correlated with the *topics* of the documents, which implies high $I(T_d; C)$ values, where C is a random variable corresponding to these (hidden) topics (see [26] for a detailed discussion). Although C is not explicitly present in our setting, clearly maximizing the information about W also improves (at least approximately) the information about C. In the following we provide some theoretical analysis to motivate these findings.

B.1 Relating maximizing information to maximizing precision

We assume the following setting. We are given a set of objects $x \in \mathcal{X}$ which are represented as conditional distributions $p(y \mid x)$. The *true* (unknown) classification of these objects induces a partition of $\mathcal{X}$ into K disjoint classes where each class is characterized through a distribution $p(y \mid c),\ c \in \mathcal{C}$. Note that this setting is consistent with the generative model discussed in Appendix A. Denoting the class of some specific $x \in \mathcal{X}$ by $c(x)$ we assume the following (strong) asymptotic assumption: $p(y \mid x) = p(y \mid c(x))\ \forall x \in \mathcal{X}$.

In the context of document classification $p(y \mid x)$ is typically estimated as the relative frequencies of the words $y \in \mathcal{Y}$ in some document x while $p(y \mid c(x))$ represents the relative frequencies of the words over all the documents that belong to the class $c(x)$. Therefore, the violation of this assumption becomes less severe as the sample size for $p(y \mid x)$ (i.e., the length of the document x) is increased.

Using the labeling scheme described in Section 4.5.2, for any given partition T, the micro-averaged precision, $Prec(T)$, is well defined. In particular, if we denote by T^* the partition which is perfectly correlated with the true classes, then clearly $Prec(T^*) = 1$.

Note that every partition T defines a set of "hard" membership probabilities $p(t \mid x)$. These probabilities in turn, defines through Eqs. (2.12) (using the IB Markovian independence relation) the set of centroid distributions $p(y \mid t)$ and prior distribution $p(t)$. Therefore, for any partition T, the mutual information $I(T; Y)$ is well defined. Under the above assumption we get:

Proposition B.1.1: $I(T^*; Y) > I(T; Y)$ *for any partition* $T \neq T^*$ *such that* $|\mathcal{T}| = K$.

Thus, the "true" partition T^*, maximizes the relevant information, and by definition the precision.

Proof: Let T be some ("hard") partition of $\mathcal{X}$. The conditional entropy about Y can be written as:

$$\begin{aligned} H(Y \mid T) &= -\sum_{t,y} p(y,t) \log p(y \mid t) \\ &= -\sum_{x,t,y} p(x,y,t) \log p(y \mid t) \\ &= -\sum_{x,t,y} p(x,y)p(t \mid x) \log p(y \mid t) \\ &= -\sum_{x,y} p(x,y) \log p(y \mid t(x)) \ , \end{aligned}$$

where in the third step we used the IB Markovian relation and in the last step we used the fact that T induces a "hard" partition.

Now, let T be some partition which is different from T^* (that is the difference between the two partitions is more than just trivial permutations). Then,

$$\begin{aligned} I(T^*;Y) - I(T;Y) &= H(Y \mid T) - H(Y \mid T^*) \\ &= \sum_{x,y} p(x,y) \log p(y \mid t^*(x)) - \sum_{x,y} p(x,y) \log p(y \mid t(x)) \\ &= \sum_{x,y} p(x,y) \log \frac{p(y \mid t^*(x))}{p(y \mid t(x))} \ . \end{aligned} \tag{B.1}$$

However, since T^* is the true ("hard") partition, then for $t_k \in T^*$, using the IB Markovian relation we have

$$\begin{aligned} p(y,t_k) &= \sum_{x} p(x,y,t_k) \\ &= \sum_{x} p(x)p(y \mid x)p(t_k \mid x) \\ &= \sum_{x \in t_k} p(x)p(y \mid x) \\ &= \sum_{x \in t_k} p(x)p(y \mid c_k) \\ &= p(y \mid c_k) \sum_{x \in t_k} p(x) \\ &= p(y \mid c_k)p(t_k) \ . \end{aligned}$$

That is, for any $t_k \in T^*$, $p(y \mid t_k) = p(y \mid c_k)$, where c_k is the corresponding class in $\mathcal{C}$. Setting this in Eq. (B.1) we obtain

$$I(T^*;Y) - I(T;Y) = \sum_{x,y} p(x,y) \log \frac{p(y \mid c(x))}{p(y \mid t(x))} \ .$$

However, using again our asymptotic assumption we know that $p(y \mid c(x)) = p(y \mid x), \ \forall x \in \mathcal{X}$, thus we obtain

$$\begin{aligned} I(T^*;Y) - I(T;Y) &= \sum_{x,y} p(x)p(y \mid x) \log \frac{p(y \mid x)}{p(y \mid t(x))} \\ &= \sum_{x} p(x) D_{KL}[p(y \mid x) \| p(y \mid t(x))] \ \geq \ 0 \ . \end{aligned} \tag{B.2}$$

Note that equality holds if and only if $p(y \mid t(x)) = p(y \mid x),\ \forall x \in \mathcal{X}$, which implies $T \equiv T^*$. Thus, for T which is different from T^* we have $I(T^*;Y) > I(T;Y)$, as required. ▮

Nonetheless, this proposition refers only to the perfect (true) partition and does not provides insight about the information preserved by other partitions. In the following, we show that a partition is (on the average) more "similar" to the true partition if and only if it is also more informative about Y. We define the distortion of some partition T with respect to the true classification by $D(T) \equiv E_{p(x)}[D_{KL}[p(y \mid c(x)) \| p(y \mid t(x))]]$. Based on our asymptotic assumption, we then get:

Proposition B.1.2: $D(T^{(1)}) \leq D(T^{(2)}) \iff I(T^{(1)};Y) \geq I(T^{(2)};Y)$

Hence, roughly speaking, seeking partitions which are more similar to the true classification is equivalent to seeking partitions that are more informative about the feature space Y.

Proof: Using Eq. (B.2) and our asymptotic assumption we have

$$\begin{aligned} I(T^*;Y) - I(T;Y) &= \sum_x p(x) D_{KL}[p(y \mid c(x)) \| p(y \mid t(x))] \\ &= D(T)\ . \end{aligned}$$

Therefore, for any two "hard" partitions, $T^{(1)}$ and $T^{(2)}$ we obtain

$$\begin{aligned} I(T^{(1)};Y) - I(T^{(2)};Y) &= I(T^*;Y) - I(T^{(2)};Y) - (I(T^*;Y) - I(T^{(1)};Y)) \\ &= D(T^{(2)}) - D(T^{(1)})\ , \end{aligned}$$

as required. ▮

A natural question is whether we can relax our asymptotic assumption while still proving the above statements, which we leave for future research.

Appendix C

Proofs for Part II

In this appendix we sketch the proofs of the theorems and propositions mentioned throughout Part II. The order of the proofs follows the order of appearance in the text.

C.1 Proofs for Section 2.1.1

Proof of Theorem 2.1.1:
We consider the functional

$$\begin{aligned}\tilde{\mathcal{F}}(p(t \mid x)) &= I(T;X) + \beta \, \langle \, d(x,t) \, \rangle_{p(x)p(t|x)} \\ &+ \sum_x \lambda(x) \sum_t p(t \mid x) \; ,\end{aligned}$$

where the last term corresponds to the normalization constraints. Recall that

$$I(T;X) = \sum_{x,t} p(x)p(t \mid x) \log \frac{p(t \mid x)}{p(t)} \; , \tag{C.1}$$

where $p(t)$ is the marginal distribution of $p(x)p(t \mid x)$, that is

$$p(t) = \sum_x p(x)p(t \mid x) \; . \tag{C.2}$$

Additionally, recall that

$$\langle \, d(x,t) \, \rangle_{p(x)p(t|x)} = \sum_{x,t} p(x)p(t \mid x)d(x,t) \; . \tag{C.3}$$

Therefore, we can express $\tilde{F}$ in terms of $p(x)$ (which is the constant source statistics) and $p(t \mid x)$ (which are the free parameters). Assuming that $d(x,t)$ is independent of $p(t \mid x)$ (which is true for rate distortion, but *not* true for the IB problem), we can differentiate with respect to $p(t \mid x)$, and get

$$\begin{aligned}\frac{\delta \tilde{\mathcal{F}}}{\delta p(t \mid x)} &= p(x) \log \frac{p(t \mid x)}{p(t)} + p(x) \\ &- \sum_{x'} p(x')p(t \mid x') \frac{1}{p(t)} p(x) \\ &+ \beta \cdot p(x)d(x,t) + \lambda(x) = 0 \; .\end{aligned}$$

Using Eq. (C.2) and simple algebra, we obtain

$$p(t \mid x) = \frac{p(t)}{Z(\beta, x)} e^{-\beta d(x,t)} \ , \tag{C.4}$$

where $Z(\beta, x)$ does not depend on t. Since the normalization constraints must hold it follows that $Z(\beta, x)$ is the normalization (partition) function

$$Z(\beta, x) = \sum_t p(t) e^{-\beta d(x,t)} \ , \tag{C.5}$$

as required.

To verify Eq. (2.6) note that when varying the (normalized) distributions $p(t \mid x)$ the variations $\delta I(T;X)$ and $\delta \langle d(x,t) \rangle_{p(x)p(t|x)}$ are linked through

$$\delta \mathcal{F} = \delta I(T;X) + \beta \cdot \delta \langle d(x,t) \rangle_{p(x)p(t|x)} = 0 \ , \tag{C.6}$$

from which Eq. (2.6) follows. ∎

Proof of Proposition 2.1.2:
We repeat the proof from [20], page 365.

$$\begin{array}{rcl} D_{KL}[p(x)p(t \mid x) \| p(x)p(t)] & - & D_{KL}[p(x)p(t \mid x) \| p(x)p^*(t)] \\ & = & \displaystyle\sum_{x,t} p(x)p(t \mid x) \log \frac{p(x)p(t \mid x)}{p(x)p(t)} \\ & - & \displaystyle\sum_{x,t} p(x)p(t \mid x) \log \frac{p(x)p(t \mid x)}{p(x)p^*(t)} \\ & = & \displaystyle\sum_{x,t} p(x)p(t \mid x) \log \frac{p^*(t)}{p(t)} \\ & = & \displaystyle\sum_{t} p^*(t) \log \frac{p^*(t)}{p(t)} \\ & = & D_{KL}[p^*(t) \| p(t)] \ \geq \ 0 \ . \end{array}$$

∎

C.2 Proofs for Section 2.3

Proof of Proposition 2.3.2:
Consider Definition 2.3.1 of the relevance-compression function, $\hat{R}(\hat{D})$. As $\hat{D}$ increases, the set of conditional distributions $p(t \mid x)$ for which $I(T;Y) \geq \hat{D}$ can only *decrease*. Hence, as $\hat{D}$ increases, $\hat{R}(\hat{D})$ becomes the minimum of $I(T;X)$ over decreasingly smaller sets. As a result, $\hat{R}(\hat{D})$ can only increase with $\hat{D}$, i.e., it is a monotonic non-decreasing function of $\hat{D}$.

As in Eq. (C.6) we note that when varying the (normalized) distributions $p(t \mid x)$ the variations $\delta I(T;X)$ and $\delta I(T;Y)$ are linked through

$$\delta \mathcal{L} = \delta I(T;X) - \beta \delta I(T;Y) = 0 \ , \tag{C.7}$$

from which Eq. (2.15) follows.

As a result we see that $\hat{R}(\hat{D})$ is a monotonic non-decreasing function with a monotonically decreasing slope, and as such it is a concave function of $\hat{D}$. ∎

C.3 Proofs for Section 2.4

Proof of Theorem 2.4.1:
We need to consider the functional $\tilde{\mathcal{L}} = \mathcal{L} + \sum_x \lambda(x) \sum_t p(t \mid x)$, where the last term corresponds to the normalization constraints. Writing $\tilde{\mathcal{L}}$ explicitly we have

$$\begin{aligned} \tilde{\mathcal{L}} &= \sum_{x,t} p(x)p(t \mid x) \log \frac{p(t \mid x)}{p(t)} \\ &- \beta \sum_{t,y} p(t,y) \log \frac{p(t,y)}{p(t)p(y)} \\ &+ \sum_x \lambda(x) \sum_t p(t \mid x) \ , \end{aligned}$$

where $p(t), \ p(t,y)$ are defined through the IB Markovian relation $T \leftrightarrow X \leftrightarrow Y$ (see Eqs. (2.12)). That is,

$$\begin{cases} p(t) = \sum_x p(x)p(t \mid x) \\ \\ p(t,y) = \sum_x p(x,y)p(t \mid x) \ . \end{cases} \tag{C.8}$$

Therefore, differentiating with respect to some $p(t \mid x)$ we obtain

$$\begin{cases} \frac{\delta p(t)}{\delta p(t|x)} = p(x) \\ \\ \frac{\delta p(t,y)}{\delta p(t|x)} = p(x,y) \ . \end{cases} \tag{C.9}$$

Using these partial derivatives we can now differentiate $\tilde{\mathcal{L}}$.

$$\begin{aligned} \frac{\delta \tilde{\mathcal{L}}}{\delta p(t \mid x)} &= p(x)(\log p(t \mid x) + 1) - p(x) \log p(t) - p(x) \\ &- \beta \sum_y p(x,y) \log p(y \mid t) + \beta \sum_y p(x,y) \log p(t) \\ &+ \tilde{\lambda}(\beta, x) \ = \ 0 \ , \end{aligned}$$

where we absorb in $\tilde{\lambda}(\beta, x)$ terms that does not depend on t. Dividing by $p(x)$ and rearranging we have

$$\log p(t \mid x) = \log p(t) - \beta \sum_y p(y \mid x) \log \frac{1}{p(y \mid t)} - \tilde{\lambda}(\beta, x) \ . \tag{C.10}$$

To get the KL form, we add and subtract $\beta \sum_y p(y \mid x) \log p(y \mid x)$ (which does not depend on t, hence can be further absorbed by $\tilde{\lambda}$), to obtain

$$\log p(t \mid x) = \log p(t) - \beta D_{KL}[p(y \mid x) \| p(y \mid t)] - \tilde{\lambda}(\beta, x) \ . \tag{C.11}$$

Taking the exponent and using again the normalization constraints, we have

$$p(t \mid x) = \frac{p(t)}{Z(\beta, x)} e^{-\beta D_{KL}[p(y|x) \| p(y|t)]} \ , \tag{C.12}$$

where $Z(\beta, x)$ guarantees the normalization, as required. ▮

C.4 Proofs for Section 3.3

Proof of Proposition 3.3.1:

$$\begin{aligned}
p(\bar{t}) &= \sum_x p(x)p(\bar{t} \mid x) \\
&= \sum_x p(x)(p(t_i \mid x) + p(t_j \mid x)) \\
&= \sum_x p(x)p(t_i \mid x) + \sum_x p(x)p(t_j \mid x) \\
&= p(t_i) + p(t_j) \ .
\end{aligned}$$

$$\begin{aligned}
p(y, \bar{t}) &= \sum_x p(x,y)p(\bar{t} \mid x) \\
&= \sum_x p(x,y)(p(t_i \mid x) + p(t_j \mid x)) \\
&= \sum_x p(x,y)p(t_i \mid x) + \sum_x p(x,y)p(t_j \mid x) \\
&= p(y, t_i) + p(y, t_j) \ .
\end{aligned}$$

Therefore,

$$p(y \mid \bar{t}) = \frac{p(t_i)}{p(\bar{t})} p(y \mid t_i) + \frac{p(t_j)}{p(\bar{t})} p(y \mid t_j) \ . \tag{C.13}$$

▮

C.5 Proofs for Section 3.3.1

Proof of Proposition 3.3.2:
Let T^{bef} and T^{aft} denote the random variables that correspond to T, before and after the merger, respectively. Thus, the corresponding values of $\mathcal{L}_{max}$ are calculated based on T^{bef} and T^{aft}. The merger cost is then given by,

$$\begin{aligned}
\Delta\mathcal{L}_{max}(t_i, t_j) &= \mathcal{L}_{max}^{bef} - \mathcal{L}_{max}^{aft} \\
&= I(T^{bef}; Y) - I(T^{aft}; Y) - \beta^{-1}(I(T^{bef}; X) - I(T^{aft}; X)) \\
&\equiv \Delta I_2 - \beta^{-1} \Delta I_1 \ .
\end{aligned}$$

We first handle the first term.

$$\begin{aligned}
\Delta I_2 &= p(t_i) \sum_y p(y \mid t_i) \log \frac{p(y \mid t_i)}{p(y)} + p(t_j) \sum_y p(y \mid t_j) \log \frac{p(y \mid t_j)}{p(y)} \\
&- p(\bar{t}) \sum_y p(y \mid \bar{t}) \log \frac{p(y \mid \bar{t})}{p(y)} \ .
\end{aligned}$$

Using Proposition 3.3.1 we obtain

$$\begin{aligned}
\Delta I_2 &= p(t_i)\sum_y p(y \mid t_i)\log\frac{p(y \mid t_i)}{p(y)} + p(t_j)\sum_y p(y \mid t_j)\log\frac{p(y \mid t_j)}{p(y)} \\
&- p(t_i)\sum_y [\pi_i p(y \mid t_i) + \pi_j p(y \mid t_j)]\log\frac{p(y \mid \bar{t})}{p(y)} \\
&- p(t_j)\sum_y [\pi_i p(y \mid t_i) + \pi_j p(y \mid t_j)]\log\frac{p(y \mid \bar{t})}{p(y)} \\
&= p(t_i)\sum_y p(y \mid t_i)\log\frac{p(y \mid t_i)}{p(y)} + p(t_j)\sum_y p(y \mid t_j)\log\frac{p(y \mid t_j)}{p(y)} \\
&- \pi_i p(t_i)\sum_y p(y \mid t_i)\log\frac{p(y \mid \bar{t})}{p(y)} - \pi_i p(t_j)\sum_y p(y \mid t_i)\log\frac{p(y \mid \bar{t})}{p(y)} \\
&- \pi_j p(t_i)\sum_y p(y \mid t_j)\log\frac{p(y \mid \bar{t})}{p(y)} - \pi_j p(t_j)\sum_y p(y \mid t_j)\log\frac{p(y \mid \bar{t})}{p(y)} \; .
\end{aligned}$$

Using $\pi_i p(t_i) + \pi_i p(t_j) = p(t_i)$ and similarly for π_j we have

$$\begin{aligned}
\Delta I_2 &= p(t_i)\sum_y p(y \mid t_i)\log\frac{p(y \mid t_i)}{p(y)} + p(t_j)\sum_y p(y \mid t_j)\log\frac{p(y \mid t_j)}{p(y)} \\
&- p(t_i)\sum_y p(y \mid t_i)\log\frac{p(y \mid \bar{t})}{p(y)} - p(t_j)\sum_y p(y \mid t_j)\log\frac{p(y \mid \bar{t})}{p(y)} \\
&= p(t_i)\sum_y p(y \mid t_i)\log\frac{p(y \mid t_i)}{p(y \mid \bar{t})} + p(t_j)\sum_y p(y \mid t_j)\log\frac{p(y \mid t_j)}{p(y \mid \bar{t})} \\
&= p(t_i)D_{KL}[p(y \mid t_i)\|p(y \mid \bar{t})] + p(t_j)D_{KL}[p(y \mid t_j)\|p(y \mid \bar{t})] \\
&= p(\bar{t}) \cdot [\pi_i D_{KL}[p(y \mid t_i)\|p(y \mid \bar{t})] + \pi_j D_{KL}[p(y \mid t_j)\|p(y \mid \bar{t})] \\
&= p(\bar{t}) \cdot JS_{\Pi}[p(y \mid t_i), p(y \mid t_j)] \; .
\end{aligned}$$

Similar analysis yields $\Delta I_1 = p(\bar{t}) \cdot JS_{\Pi}[p(x \mid t_i), p(x \mid t_j)]$, as required. ▮

Appendix D

Proofs for Part III

In this appendix we sketch the proofs of the theorems and propositions mentioned throughout Part III. The order of the proofs follows the order of appearance in the text.

D.1 Proofs for Section 7.1

Proof of Proposition 7.1.1:
Using the multi-information definition in 1.2.13 and the fact that $p(\mathbf{x}) \models G$ we get

$$\begin{aligned}
\mathcal{I}(\mathbf{X}) &= E_p[\log \frac{p(\mathbf{x})}{p(x_1)\ldots p(x_n)}] \\
&= E_p[\log \Pi_{i=1}^n \frac{p(x_i \mid \mathbf{pa}_{X_i}^G)}{p(x_i)}] \\
&= \sum_{i=1}^n E_p[\log \frac{p(x_i \mid \mathbf{pa}_{X_i}^G)}{p(x_i)}] \\
&= \sum_{i=1}^n I(X_i; \mathbf{Pa}_{X_i}^G) \ .
\end{aligned}$$

▮

Proof of Proposition 7.1.3:

$$\begin{aligned}
D_{KL}[p\|G] &= \min_{q\models G} E_p[\log \frac{p(x_1,\ldots,x_n)}{q(x_1,\ldots,x_n)}] \\
&= \min_{q\models G}[E_p[\log \frac{p(x_1,\ldots,x_n)}{\Pi_{i=1}^n p(x_i \mid \mathbf{pa}_{X_i}^G)}] + E_p[\log \frac{\Pi_{i=1}^n p(x_i \mid \mathbf{pa}_{X_i}^G)}{\Pi_{i=1}^n q(x_i \mid \mathbf{pa}_{X_i}^G)}]] \\
&= E_p[\log \frac{p(x_1,\ldots,x_n)}{\Pi_{i=1}^n p(x_i \mid \mathbf{pa}_{X_i}^G)}] + \min_{q\models G}[\sum_{i=1}^n \sum_{x_i,\mathbf{pa}_{X_i}^G} p(\mathbf{pa}_{X_i}^G) p(x_i \mid \mathbf{pa}_{X_i}^G) \log \frac{p(x_i \mid \mathbf{pa}_{X_i}^G)}{q(x_i \mid \mathbf{pa}_{X_i}^G)}] \\
&= E_p[\log \frac{p(x_1,\ldots,x_n)}{\Pi_{i=1}^n p(x_i \mid \mathbf{pa}_{X_i}^G)}] + \min_{q\models G}[\sum_{i=1}^n \sum_{\mathbf{pa}_{X_i}^G} p(\mathbf{pa}_{X_i}^G) D_{KL}[p(x_i \mid \mathbf{pa}_{X_i}^G)\|q(x_i \mid \mathbf{pa}_{X_i}^G)]] \ ,
\end{aligned}$$

and since the right term is non-negative and equals zero if and only if we choose $q(x_i \mid \mathbf{pa}_{X_i}^G) = p(x_i \mid \mathbf{pa}_{X_i}^G)$ we get the desired result. ▮

Proof of Proposition 7.1.4:
We use Proposition 7.1.3.

$$
\begin{aligned}
D_{KL}[p\|G] &= \min_{q \models G} E_p[\log \frac{p(x_1,\ldots,x_n)}{q(x_1,\ldots,x_n)}] \\
&= E_p[\log \frac{p(x_1,\ldots,x_n)}{\Pi_{i=1}^{n} p(x_i \mid \mathbf{pa}_{X_i}^{G})}] \\
&= E_p[\log \frac{\Pi_{i=1}^{n} p(x_i \mid x_1,\ldots,x_{i-1})}{\Pi_{i=1}^{n} p(x_i \mid \mathbf{pa}_{X_i}^{G})}] \\
&= \sum_{i=1}^{n} E_p[\log \frac{p(x_i \mid x_1,\ldots,x_n)}{p(x_i \mid \mathbf{pa}_{X_i}^{G})}] \\
&= \sum_{i=1}^{n} I(X_i; \{X_1,\ldots,X_n\} \setminus \mathbf{Pa}_{X_i}^{G} \mid \mathbf{Pa}_{X_i}^{G}) \ ,
\end{aligned}
$$

where we used the consistency of the order $X_1,\ldots,X_n$ with the order of the DAG G. To prove the second part of the proposition we note that

$$
\begin{aligned}
D_{KL}[p\|G] &= E_p[\log \frac{p(x_1,\ldots,x_n)}{\Pi_{i=1}^{n} p(x_i \mid \mathbf{pa}_{X_i}^{G})}] \\
&= E_p[\log \frac{p(x_1,\ldots,x_n)}{\Pi_{i=1}^{n} p(x_i)}] - E_p[\log \frac{\Pi_{i=1}^{n} p(x_i \mid \mathbf{pa}_{X_i}^{G})}{\Pi_{i=1}^{n} p(x_i)}] \\
&= \mathcal{I}(\mathbf{X}) - \sum_{i=1}^{n} E_p[\log \frac{p(x_i \mid \mathbf{pa}_{X_i}^{G})}{p(x_i)}] \\
&= \mathcal{I}(\mathbf{X}) - \sum_{i=1}^{n} I(X_i; \mathbf{Pa}_{X_i}^{G}) \ .
\end{aligned}
$$

▮

D.2 Proofs for Section 8.2

Proof of Proposition 8.2.1:
Assume that $X \leftrightarrow T \leftrightarrow Y$, then from Data Processing Inequality ([20], page 32) we get $I(T;Y) \geq I(X;Y)$. However, since $T \leftrightarrow X \leftrightarrow Y$ then for the same reason we get $I(T;Y) \leq I(X;Y)$, i.e., $I(T;Y) = I(X;Y)$.

Assume now that $I(T;Y) = I(X;Y)$. From the chain rule for mutual information ([20], page 22) we have

$$I(T,X;Y) = I(T;Y) + I(X;Y \mid T) = I(X;Y) + I(T;Y \mid X) \ , \tag{D.1}$$

hence, $I(X;Y \mid T) = I(T;Y \mid X)$. Using this result and the definition of conditional mutual information we get $H(Y \mid T) = H(Y \mid X)$. However, since $T \leftrightarrow X \leftrightarrow Y$ clearly $H(Y \mid X) = H(Y \mid X,T)$, therefore we obtain $H(Y \mid T) = H(Y \mid X,T)$, meaning $I(X;Y \mid T) = 0$. ▮

D.3 Proofs of Section 9.1

Proof of Theorem 9.1.1:
The basic idea is to find stationary points of $\mathcal{L}^{(1)}$ subject to the normalization constraints. Thus, we add Lagrange multipliers and use Definition 1.2.13 to get the Lagrangian

$$\tilde{\mathcal{L}}[p(\mathbf{x},\mathbf{t})] = \sum_{\ell=1}^{k} I(T_\ell;\mathbf{U}_\ell) - \beta[\sum_{i=1}^{n} I(X_i;\mathbf{V}_{X_i}) + \sum_{\ell=1}^{k} I(T_\ell;\mathbf{V}_{T_\ell})] + \sum_{\mathbf{u}_\ell} \lambda(\mathbf{u}_\ell) \sum_{t_\ell} p(t_\ell \mid \mathbf{u}_\ell) \ , \quad \text{(D.2)}$$

where we drop terms that depend only on the observed variables $\mathbf{X}$. To differentiate $\tilde{\mathcal{L}}$ with respect to a specific parameter $p(t_j \mid \mathbf{u}_j)$ we use the following two lemmas. In the proofs of these two lemmas we assume that $p(\mathbf{x},\mathbf{t}) \models G_{in}$ and that the $\mathbf{T}$ variables are all leafs in G_{in}.

Lemma D.3.1: *Under the above normalization constraints, for every event* $\mathbf{a}$ *over* $\mathbf{X} \cup \mathbf{T}$ *(that is,* $\mathbf{a}$ *is some assignment to some subset of* $\mathbf{X} \cup \mathbf{T}$*), we have*

$$\frac{\delta p(\mathbf{a})}{\delta p(t_j \mid \mathbf{u}_j)} = p(\mathbf{u}_j) p(\mathbf{a} \mid t_j, \mathbf{u}_j) \ . \quad \text{(D.3)}$$

Proof: Let $\mathbf{Z}$ denote all the random variables in $\mathbf{X} \cup \mathbf{T}$ such that their values are not set by the event $\mathbf{a}$. In the following, the notation $\sum_{\mathbf{z},\mathbf{a}} p(\mathbf{z},\mathbf{t})$ means that the sum is only over the variables in $\mathbf{Z}$ (where the others are set through $\mathbf{a}$).

$$\begin{aligned}
\frac{\delta p(\mathbf{a})}{\delta p(t_j \mid \mathbf{u}_j)} &= \frac{\delta}{\delta p(t_j \mid \mathbf{u}_j)} \sum_{\mathbf{z},\mathbf{a}} p(\mathbf{x},\mathbf{t}) \\
&= \frac{\delta}{\delta p(t_j \mid \mathbf{u}_j)} \sum_{\mathbf{z},\mathbf{a}} \Pi_{\ell=1}^{k} p(t_\ell \mid \mathbf{u}_\ell) \\
&= \sum_{\mathbf{z},\mathbf{a}} \frac{\delta}{\delta p(t_j \mid \mathbf{u}_j)} \Pi_{\ell=1}^{k} p(t_\ell \mid \mathbf{u}_\ell) \ .
\end{aligned}$$

Clearly the derivatives are nonzero only for terms in which $T_j = t_j$ and $\mathbf{U}_j = \mathbf{u}_j$. For each such term the derivative is simply $\Pi_{\ell=1,\ell\neq j}^{k} p(t_\ell \mid \mathbf{u}_\ell)$. Dividing and multiplying every such term by $p(t_j \mid \mathbf{u}_j)$ we obtain

$$\begin{aligned}
\frac{\delta p(a)}{\delta p(t_j \mid \mathbf{u}_j)} &= \frac{1}{p(t_j \mid \mathbf{u}_j)} \sum_{\mathbf{z}\setminus\{t_j,\mathbf{u}_j\},a,t_j,\mathbf{u}_j} \Pi_{\ell=1}^{k} p(t_\ell \mid \mathbf{u}_\ell) \\
&= \frac{p(a,t_j,\mathbf{u}_j)}{p(t_j \mid \mathbf{u}_j)} \\
&= p(\mathbf{u}_j) p(a \mid t_j, \mathbf{u}_j) \ .
\end{aligned}$$

▮

Using this lemma we get:

Lemma D.3.2: *For every* $Y, \mathbf{Z} \subseteq \mathbf{X} \cup \mathbf{T}$

$$\frac{\delta I(Y;\mathbf{Z})}{\delta p(t_j \mid \mathbf{u}_j)} = p(\mathbf{u}_j) \sum_{y,\mathbf{z}} [p(y,\mathbf{z} \mid t_j,\mathbf{u}_j) \log \frac{p(y \mid \mathbf{z})}{p(y)} - 1] \ . \quad \text{(D.4)}$$

Proof:

$$\begin{aligned}\frac{\delta I(Y;\mathbf{Z})}{\delta p(t_j \mid \mathbf{u}_j)} &= \sum_{\mathbf{y},\mathbf{z}} \log \frac{p(\mathbf{y} \mid \mathbf{z})}{p(\mathbf{y})} \frac{\delta}{\delta p(t_j \mid \mathbf{u}_j)} p(\mathbf{y},\mathbf{z}) \\ &+ \sum_{\mathbf{y},\mathbf{z}} \frac{\delta}{\delta p(t_j \mid \mathbf{u}_j)} p(\mathbf{y},\mathbf{z}) \\ &- \sum_{\mathbf{y},\mathbf{z}} p(\mathbf{z} \mid \mathbf{y}) \frac{\delta}{\delta p(t_j \mid \mathbf{u}_j)} p(\mathbf{y}) \\ &- \sum_{\mathbf{y},\mathbf{z}} p(\mathbf{y} \mid \mathbf{z}) \frac{\delta}{\delta p(t_j \mid \mathbf{u}_j)} p(\mathbf{z})\ .\end{aligned}$$

Applying Lemma D.3.1 for each of these derivatives we get the desired result. ▮

We now can differentiate each mutual information term that appears in $\tilde{\mathcal{L}}$ of Eq. (D.2). Note that we can ignore terms that do not depend on the value of T_j since these are constants with respect to $p(t_j \mid \mathbf{u}_j)$. Therefore, by taking the derivative and equating to zero we get:

$$\begin{aligned}\log p(t_j \mid \mathbf{u}_j) &= \log p(t_j) \\ & -\beta[\sum_{i:T_j \in \mathbf{V}_{X_i}} \sum_{\mathbf{v}_{X_i}^{-j}, x_i} p(\mathbf{v}_{X_i}^{-j} \mid \mathbf{u}_j) p(x_i \mid \mathbf{v}_{X_i}^{-j}, \mathbf{u}_j) \log \frac{p(x_i)}{p(x_i \mid \mathbf{v}_{X_i}^{-j}, t_j)} \\ & - \sum_{\ell:T_j \in \mathbf{V}_{T_\ell}} \sum_{\mathbf{v}_{T_\ell}^{-j}, t_\ell} p(\mathbf{v}_{T_\ell}^{-j} \mid \mathbf{u}_j) p(t_\ell \mid \mathbf{v}_{T_\ell}^{-j}, \mathbf{u}_j) \log \frac{p(t_\ell)}{p(t_\ell \mid \mathbf{v}_{T_\ell}^{-j}, t_j)} \\ & - \sum_{\mathbf{v}_{T_j}} p(\mathbf{v}_{T_j} \mid \mathbf{u}_j) \log \frac{p(\mathbf{v}_{T_j})}{p(\mathbf{v}_{T_j} \mid t_j)}] + c(\mathbf{u}_j)\ , \end{aligned} \qquad \text{(D.5)}$$

where $c(\mathbf{u}_j)$ is a term that depends only on $\mathbf{u}_j$. To get the desired KL form we add and subtract

$$\sum_{\mathbf{v}_{X_i}^{-j}, X_i} p(\mathbf{v}_{X_i}^{-j} \mid \mathbf{u}_j) p(x_i \mid \mathbf{v}_{X_i}^{-j}, \mathbf{u}_j) \log \frac{p(x_i \mid \mathbf{v}_{X_i}^{-j}, \mathbf{u}_j)}{p(x_i)}\ , \qquad \text{(D.6)}$$

for every term in the first outside summation. Note again that this is possible since we can absorb in $c(\mathbf{u}_j)$ every expression that depends only on $\mathbf{u}_j$. A Similar transformation applies to the other two summations in the right hand side of Eq. (D.5). Hence, we end up with

$$\begin{aligned}\log p(t_j \mid \mathbf{u}_j) &= \log p(t_j) \\ & -\beta \cdot [\sum_{i:T_j \in \mathbf{V}_{X_i}} E_{p(\cdot \mid \mathbf{u}_j)}[D_{KL}[p(x_i \mid \mathbf{v}_{X_i}^{-j}, \mathbf{u}_j) \| p(x_i \mid \mathbf{v}_{X_i}^{-j}, t_j)]] \\ & - \sum_{\ell:T_j \in \mathbf{V}_{T_\ell}} E_{p(\cdot \mid \mathbf{u}_j)}[D_{KL}[p(t_\ell \mid \mathbf{v}_{T_\ell}^{-j}, \mathbf{u}_j) \| p(t_\ell \mid \mathbf{v}_{T_\ell}^{-j}, t_j)]] \\ & - D_{KL}[p(\mathbf{v}_{T_j} \mid \mathbf{u}_j) \| p(\mathbf{v}_{T_j} \mid t_j)]\ ] + c(\mathbf{u}_j)\ . \end{aligned} \qquad \text{(D.7)}$$

Finally, taking the exponent and applying the normalization constraints for each distribution $p(t_j \mid \mathbf{u}_j)$ completes the proof. ▮

D.4 Proofs of Section 10.1

Proof of Theorem 10.1.1:
We start by introducing the following auxiliary functional:

$$F \equiv \sum_{j=1}^{k} F_j \equiv -\sum_{j=1}^{k} \sum_{t_j} \sum_{\mathbf{u}_j} p(\mathbf{u}_j) p(t_j \mid \mathbf{u}_j) \log Z_{T_j}(\mathbf{u}_j, \beta) \;, \tag{D.8}$$

where, as before, $Z_{T_j}(\mathbf{u}_j, \beta)$ is the normalization (partition) function of $p(t_j \mid \mathbf{u}_j)$. In other words, F is (minus) the averaged log of all the partition functions. The general idea of the proof is similar to the proof of Theorem 3.1.1. Specifically, we show that for every j, the updates defined by the multivariate iIB algorithm can only reduce F (or more precisely, reduce F_j). Since F is shown to be lower-bounded, we are guaranteed to converge to a self-consistent solution.

Lemma D.4.1: *F is non-negative and strictly convex with respect to each of its arguments.*

Proof: Using Eq. (9.1) we find that

$$\begin{aligned} F &= \sum_{j=1}^{k} F_j & \text{(D.9)} \\ &= \sum_{j=1}^{k} \sum_{t_j} \sum_{\mathbf{u}_j} p(\mathbf{u}_j) p(t_j \mid \mathbf{u}_j) \log \frac{p(t_j \mid \mathbf{u}_j)}{p(t_i)} & \text{(D.10)} \\ &+ \beta \sum_{j=1}^{k} \sum_{t_j} \sum_{\mathbf{u}_j} p(\mathbf{u}_j) p(t_j \mid \mathbf{u}_j) d(t_j, \mathbf{u}_j) \;. & \text{(D.11)} \end{aligned}$$

Therefore, F is a sum of KL divergences, and in particular non negative. Moreover, since the KL is strictly convex with respect to each of its arguments (which results from Log sum inequality [20]), F is convex *independently* in each argument (as a sum of convex functions). ▮.

Note that after updating $p(t_j)$ by the multivariate iIB algorithm, $p(t_j)$ becomes exactly the marginal of the joint distribution $p(\mathbf{u}_j)p(t_j \mid \mathbf{u}_j)$. Therefore, after completing the updates for all $j = 1 : k$, the first term in F corresponds to $\mathcal{I}^{G_{in}}$. Moreover, at any stage, even if $p(t_j)$ is not set to be the appropriate marginal distribution, this ("compression") term is always lower bounded by $\mathcal{I}^{G_{in}}$ (see, e.g., [20], page 365).

Lemma D.4.2: *If the multivariate iIB update steps for some T_j changes any of the involved distributions, F is reduced.*

Proof: First, let us write explicitly all the iIB update steps for some T_j. The updates of $p(t_j \mid \mathbf{u}_j)$ and $p(t_j)$ are already described in Figure 10.1. The additional updates are as follows. For every i such that $T_j \in \mathbf{V}_{X_i}$ we update:

$$p^{(m+1)}(x_i \mid \mathbf{v}_{X_i}^{-j}, t_j) \leftarrow \frac{1}{\lambda(\mathbf{v}_{X_i}^{-j}, t_j)} \sum_{\mathbf{u}_j} p^{(m+1)}(t_j \mid \mathbf{u}_j) p^{(m)}(\mathbf{u}_j, \mathbf{v}_{X_i}^{-j}, x_i) \;, \tag{D.12}$$

where $\lambda(\mathbf{v}_{X_i}^{-j}, t_j)$ guarantees the proper normalization. Second, for every ℓ such that $T_j \in \mathbf{V}_{T_\ell}$ we update:

$$p^{(m+1)}(t_\ell \mid \mathbf{v}_{T_\ell}^{-j}, t_j) \leftarrow \frac{1}{\lambda(\mathbf{v}_{T_\ell}^{-j}, t_j)} \sum_{\mathbf{u}_j} p^{(m+1)}(t_j \mid \mathbf{u}_j) p^{(m)}(\mathbf{u}_j, \mathbf{v}_{T_\ell}^{-j}, t_\ell) \;, \tag{D.13}$$

where $\lambda(\mathbf{v}_{T_\ell}^{-j}, t_j)$ guarantees the proper normalization. Lastly, if $\mathbf{v}_{T_j} \neq \emptyset$, we update:

$$p^{(m+1)}(\mathbf{v}_{T_j} \mid t_j) \leftarrow \frac{1}{\lambda(t_j)} \sum_{\mathbf{u}_j} p^{(m+1)}(t_j \mid \mathbf{u}_j) p^{(m)}(\mathbf{u}_j, \mathbf{v}_{T_j}) \; , \tag{D.14}$$

where $\lambda(t_j)$ guarantees the proper normalization. We now note that the derivatives of F with respect to each of its arguments (under proper normalization constraints), provide exactly the above multivariate iIB update steps. For example, consider $\tilde{F} \equiv F + \sum_{t_j} \lambda(t_j)[\sum_{\mathbf{v}_{T_j}} p(\mathbf{v}_{T_j}) - 1]$, where the second term corresponds to the normalization constraints. Taking the derivative of $\tilde{F}$ with respect to $p(\mathbf{v}_{T_j} \mid t_j)$ and equating to zero will give exactly Eq. (D.14). A similar procedure for the other arguments of F will yield exactly all the other multivariate iIB steps.

Therefore, updating by equating some derivative of F to zero (while all the other arguments of F remain constant), can only reduce F. This is simply due to the fact that F is strictly convex (independently in each argument) and all its arguments correspond to convex sets. Hence, equating some derivative of F to zero is equivalent to finding the projection of F in the corresponding convex set. This can only reduce F, or leave it unchanged, where in this case the update step has no effect. ▮

Combining the above two lemmas we see that through these updates F converges to a (local) minimum. At this point all the update steps (including Eq. (9.1)) reach a self-consistent solution. Therefore, from Theorem 9.1.1 we are at a fixed-point of $\mathcal{L}^{(1)}$, as required. ▮

D.5 Proofs for Section 10.3

Proof of Proposition 10.3.2:
We use the following notations: $\mathbf{W} = \mathbf{Z} \cap \mathbf{U}_j,\ \mathbf{Z}^{-\mathbf{W}} = \mathbf{Z} \setminus \{\mathbf{W}\},\ \mathbf{U}_j{}^{-\mathbf{W}} = \mathbf{U}_j \setminus \{\mathbf{W}\}$. Note that in principle it might be that $\mathbf{W} = \emptyset$.

$$\begin{aligned} p(\mathbf{z}, \bar{t}_j) &= p(\mathbf{z}) p(\bar{t}_j \mid \mathbf{z}) \\ &= p(\mathbf{z}) \sum_{\mathbf{u}_j{}^{-\mathbf{w}}} p(\mathbf{u}_j{}^{-\mathbf{w}} \mid \mathbf{z}) p(\bar{t}_j \mid \mathbf{z}^{-\mathbf{w}}, \mathbf{w}, \mathbf{u}_j{}^{-\mathbf{w}}) \\ &= p(\mathbf{z}) \sum_{\mathbf{u}_j{}^{-\mathbf{w}}} p(\mathbf{u}_j{}^{-\mathbf{w}} \mid \mathbf{z}) p(\bar{t}_j \mid \mathbf{u}_j) \; , \end{aligned}$$

where in the last step we used the structure of G_{in} and the fact that $\mathbf{Z}^{-\mathbf{W}} \cap \mathbf{U}_j = \emptyset$. Using Eq. (10.3) we find that

$$\begin{aligned} p(\mathbf{z}, \bar{t}_j) &= p(\mathbf{z}) \sum_{\mathbf{u}_j{}^{-\mathbf{w}}} p(\mathbf{u}_j{}^{-\mathbf{w}} \mid \mathbf{z}) (p(t_j^\ell \mid \mathbf{u}_j) + p(t_j^r \mid \mathbf{u}_j)) \\ &= p(\mathbf{z}) \sum_{\mathbf{u}_j{}^{-\mathbf{w}}} p(\mathbf{u}_j{}^{-\mathbf{w}} \mid \mathbf{z}) (p(t_j^\ell \mid \mathbf{z}^{-\mathbf{w}}, \mathbf{w}, \mathbf{u}_j{}^{-\mathbf{w}}) + p(t_j^r \mid \mathbf{z}^{-\mathbf{w}}, \mathbf{w}, \mathbf{u}_j{}^{-\mathbf{w}})) \; , \end{aligned}$$

where again we used the structure of G_{in}. Since $\mathbf{Z} = \mathbf{Z}^{-\mathbf{W}} \cup \{\mathbf{W}\}$ we get

$$\begin{aligned} p(\mathbf{z}, \bar{t}_j) &= p(\mathbf{z}) \sum_{\mathbf{u}_j{}^{-\mathbf{w}}} (p(\mathbf{u}_j{}^{-\mathbf{w}}, t_j^\ell \mid \mathbf{z}) + p(\mathbf{u}_j{}^{-\mathbf{w}}, t_j^r \mid \mathbf{z})) \\ &= p(\mathbf{z}, t_j^\ell) + p(\mathbf{z}, t_j^r) \; , \end{aligned}$$

as required.

To prove the second part we first note that if $p(\mathbf{z}, \bar{t}_j) = 0$ then both sides of Eq. (10.6) are trivially equal, thus we assume that this is not the case.

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{z}, \bar{t}_j) &= \frac{p(\mathbf{y}, \mathbf{z}, \bar{t}_j)}{p(\mathbf{z}, \bar{t}_j)} \\ &= \frac{p(\mathbf{y}, \mathbf{z}, t_j^\ell) + p(\mathbf{y}, \mathbf{z}, t_j^r)}{p(\mathbf{z}, \bar{t}_j)} \\ &= \frac{p(t_j^\ell \mid \mathbf{z})}{p(\bar{t}_j \mid \mathbf{z})} p(\mathbf{y} \mid \mathbf{z}, t_j^\ell) + \frac{p(t_j^r \mid \mathbf{z})}{p(\bar{t}_j \mid \mathbf{z})} p(\mathbf{y} \mid \mathbf{z}, t_j^r) \ , \end{aligned}$$

hence from Definition 10.3.1 we get the desired form. ▮

D.6 Proofs for Section 10.3.1

Proof of Theorem 10.3.3:
We first prove a simple Lemma. Recall that we denote by T_j^{bef}, T_j^{aft} the random variables that correspond to T_j before and after the merger, respectively. Let $\mathbf{V} = \mathbf{V}^{-j} \cup T_j$ be a set of random variables that includes T_j and let $\mathbf{V}^{bef} = \mathbf{V}^{-j} \cup T_j^{bef}$ and similarly for $\mathbf{V}^{aft}$. Let $\mathbf{Y}$ be a set of random variables such that $T_j \notin \mathbf{Y}$. Using these notations we have:

Lemma D.6.1: *The reduction of the mutual information $I(\mathbf{Y}; \mathbf{V})$ due to the merger $\{t_j^\ell, t_j^r\} \Rightarrow \bar{t}_j$ is given by*

$$\begin{aligned} \Delta I(\mathbf{Y}; \mathbf{V}) &\equiv I(\mathbf{Y}; \mathbf{V}^{bef}) - I(\mathbf{Y}; \mathbf{V}^{afT}) \\ &= p(\bar{t}_j) \cdot E_{p(\cdot|\bar{t}_j)}[\ JS_{\Pi_{\mathbf{v}^{-j}}}[p(\mathbf{y} \mid t_j^\ell, \mathbf{v}^{-j}), p(\mathbf{y} \mid t_j^r, \mathbf{v}^{-j})]\ ] \ . \end{aligned}$$

Proof: Using the chain rule for mutual information ([20], page 22) we get

$$\begin{aligned} \Delta I(\mathbf{Y}; \mathbf{V}) &= I(\mathbf{V}^{-j}; \mathbf{Y}) + I(T_j^{bef}; \mathbf{Y} \mid \mathbf{V}^{-j}) - I(\mathbf{V}^{-j}; \mathbf{Y}) - I(T_j^{aft}; \mathbf{Y} \mid \mathbf{V}^{-j}) \\ &= I(T_j^{bef}; \mathbf{Y} \mid \mathbf{V}^{-j}) - I(T_j^{aft}; \mathbf{Y} \mid \mathbf{V}^{-j}) \ . \end{aligned}$$

From Eq. (10.3), we find that

$$\Delta I(\mathbf{Y}; \mathbf{V}) = \sum_{\mathbf{v}^{-j}} p(\mathbf{v}^{-j}) \Delta I(\mathbf{v}^{-j}) \ ,$$

where we used the notation

$$\begin{aligned} \Delta I(\mathbf{v}^{-j}) = &\sum_{\mathbf{y}} p(t_j^\ell, \mathbf{y} \mid \mathbf{v}^{-j}) \log \frac{p(\mathbf{y} \mid t_j^\ell, \mathbf{v}^{-j})}{p(\mathbf{y} \mid \mathbf{v}^{-j})} \\ &+ \sum_{\mathbf{y}} p(t_j^r, \mathbf{y} \mid \mathbf{v}^{-j}) \log \frac{p(\mathbf{y} \mid t_j^r, \mathbf{v}^{-i})}{p(\mathbf{y} \mid \mathbf{v}^{-j})} \\ &- \sum_{\mathbf{y}} p(\bar{t}_j, \mathbf{y} \mid \mathbf{v}^{-j}) \log \frac{p(\mathbf{y} \mid \bar{t}_j, \mathbf{v}^{-i})}{p(\mathbf{y} \mid \mathbf{v}^{-j})} \ . \end{aligned}$$

Using Proposition 10.3.2 (with $\mathbf{Z} = \mathbf{Y} \cup \mathbf{V}^{-j}$) we obtain

$$p(\bar{t}_j, \mathbf{y} \mid \mathbf{v}^{-j}) = p(t_j^\ell, \mathbf{y} \mid \mathbf{v}^{-j}) + p(t_j^r, \mathbf{y} \mid \mathbf{v}^{-j}) \ .$$

Setting this in the previous equation we get,

$$\begin{aligned}
\Delta I(\mathbf{v}^{-j}) &= \sum_{\mathbf{y}} p(t_j^\ell, \mathbf{y} \mid \mathbf{v}^{-j}) \log \frac{p(\mathbf{y} \mid t_j^\ell, \mathbf{v}^{-j})}{p(\mathbf{y} \mid \bar{t}_j, \mathbf{v}^{-j})} + \sum_{\mathbf{y}} p(t_j^r, \mathbf{y} \mid \mathbf{v}^{-j}) \log \frac{p(\mathbf{y} \mid t_j^r, \mathbf{v}^{-j})}{p(\mathbf{y} \mid \bar{t}_j, \mathbf{v}^{-j})} \\
&= p(\bar{t}_j \mid \mathbf{v}^{-j}) \cdot \pi_{\ell, \mathbf{v}^{-j}} \sum_{\mathbf{y}} p(\mathbf{y} \mid t_j^\ell, \mathbf{v}^{-j}) \log \frac{p(\mathbf{y} \mid t_j^\ell, \mathbf{v}^{-j})}{p(\mathbf{y} \mid \bar{t}_j, \mathbf{v}^{-j})} \\
&+ p(\bar{t}_j \mid \mathbf{v}^{-j}) \cdot \pi_{r, \mathbf{v}^{-j}} \sum_{\mathbf{y}} p(\mathbf{y} \mid t_j^r, \mathbf{v}^{-j}) \log \frac{p(\mathbf{y} \mid t_j^r, \mathbf{v}^{-j})}{p(\mathbf{y} \mid \bar{t}_j, \mathbf{v}^{-j})} \, .
\end{aligned}$$

However, using again Proposition 10.3.2 we see that

$$p(\mathbf{y} \mid \bar{t}_j, \mathbf{v}^{-j}) = \pi_{\ell, \mathbf{v}^{-j}} \cdot p(\mathbf{y} \mid t_j^\ell, \mathbf{v}^{-j}) + \pi_{r, \mathbf{v}^{-j}} \cdot p(\mathbf{y} \mid t_j^r, \mathbf{v}^{-j}) \, .$$

Therefore, using the JS definition in 1.2.17 we get,

$$\Delta I(\mathbf{v}^{-j}) = p(\bar{t}_j \mid \mathbf{v}^{-j}) \cdot JS_{\Pi_{\mathbf{v}^{-j}}}[p(\mathbf{y} \mid t_j^\ell, \mathbf{v}^{-j}), p(\mathbf{y} \mid t_j^r, \mathbf{v}^{-j})] \, .$$

Setting this back in the expression for $\Delta I(\mathbf{Y}; \mathbf{V})$ we get,

$$\begin{aligned}
\Delta I(\mathbf{Y}; \mathbf{V}) &= \sum_{\mathbf{v}^{-j}} p(\mathbf{v}^{-j}) p(\bar{t}_j \mid \mathbf{v}^{-j}) \cdot JS_{\Pi_{\mathbf{v}^{-j}}}[p(\mathbf{y} \mid t_j^\ell, \mathbf{v}^{-j}), p(\mathbf{y} \mid t_j^r, \mathbf{v}^{-j})] \\
&= p(\bar{t}_j) \cdot E_{p(\cdot|\bar{t}_j)}[\, JS_{\Pi_{\mathbf{v}^{-j}}}[p(\mathbf{y} \mid t_j^\ell, \mathbf{v}^{-j}), p(\mathbf{y} \mid t_j^r, \mathbf{v}^{-j})]\,] \, .
\end{aligned}$$

▮

Using this Lemma we now prove the theorem. Note that the only information terms in $\mathcal{L} = \mathcal{I}^{G_{out}} - \beta^{-1}\mathcal{I}^{G_{in}}$ that change due to a merger in T_j are those that involve T_j. Therefore

$$\Delta \mathcal{L}(t_j^\ell, t_j^r) = \sum_{i: T_j \in \mathbf{V}_{X_i}} \Delta I(X_i; \mathbf{V}_{X_i}) + \sum_{\ell: T_j \in \mathbf{V}_{T_\ell}} \Delta I(T_\ell; \mathbf{V}_{T_\ell}) + \Delta I(T_j; \mathbf{V}_{T_j}) - \beta^{-1} \Delta I(T_j; \mathbf{U}_j) \, . \quad \text{(D.15)}$$

Applying Lemma D.6.1 for each of these information terms we get the desired form. ▮

Proof of Proposition 10.3.4:
We ask whether performing the merger $\{t_j^\ell, t_j^r\} \Rightarrow \bar{t}_j$ changes the cost of some other possible merger, $\{t_s^\ell, t_s^r\} \Rightarrow \bar{t}_s$. Let

$$\Delta \mathcal{L}(t_s^\ell, t_s^r) = p(\bar{t}_s) \cdot [\bar{d}_1 + \bar{d}_2 + \bar{d}_3 - \bar{d}_4] \, , \quad \text{(D.16)}$$

where using Theorem 10.3.3 we have

$$\begin{cases}
\bar{d}_1 = \sum_{i: T_s \in \mathbf{V}_{X_i}} E_{p(\cdot|\bar{t}_s)}[JS_{\Pi_{\mathbf{v}_{X_i}^{-s}}}[p(x_i \mid t_s^\ell, \mathbf{v}_{X_i}^{-s}), p(x_i \mid t_s^r, \mathbf{v}_{X_i}^{-s})]] \\
\bar{d}_2 = \sum_{\ell: T_s \in \mathbf{V}_{T_\ell}} E_{p(\cdot|\bar{t}_j)}[JS_{\Pi_{\mathbf{v}_{T_\ell}^{-s}}}[p(t_\ell \mid t_s^\ell, \mathbf{v}_{T_\ell}^{-s}), p(t_\ell \mid t_s^r, \mathbf{v}_{T_\ell}^{-s})]] \\
\bar{d}_3 = JS_{\Pi}[p(\mathbf{v}_{T_s} \mid t_s^\ell), p(\mathbf{v}_{T_s} \mid t_s^r)] \\
\bar{d}_4 = \beta^{-1} \cdot JS_{\Pi}[p(\mathbf{u}_s \mid t_s^\ell), p(\mathbf{u}_s \mid t_s^r)] \, .
\end{cases} \quad \text{(D.17)}$$

First, assume that $s \neq j$, then clearly $p(\bar{t}_s)$ is not affected by a merger in T_j. Now assume that T_s and T_j do *not* co-appear in any information term in $\mathcal{I}^{G_{out}}$. In this case, it is easy to verify that $\bar{d}_1, \ldots, \bar{d}_4$ does not change due to a merger in T_j. Consider for example the expression for $\bar{d}_1$. Due to our assumption if $T_s \in \mathbf{V}_{X_i}$ then necessarily $T_j \notin \mathbf{V}_{X_i}$, hence a merger in T_j cannot affect $\bar{d}_1$.

Therefore, we now assume that T_s and T_j co-appear in some information term in $\mathcal{I}^{G_{out}}$, but $p(\bar{t}_s, \bar{t}_j) = 0$. From this assumption and Proposition 10.3.2 it follows

$$p(t_s^\ell, t_j^\ell) + p(t_s^\ell, t_j^r) + p(t_s^r, t_j^\ell) + p(t_s^r, t_j^r) = 0 \ . \tag{D.18}$$

As a result, again we see that $\bar{d}_1, \ldots, \bar{d}_4$ does not change due to a merger in T_j. Consider again, for example the expression for $\bar{d}_1$. Assume that there exists some i such that $T_s, T_j \in \mathbf{V}_{X_i}$. For this i, the corresponding term in $\bar{d}_1$ is

$$\sum_{\mathbf{v}_{X_i}^{-s}} p(\mathbf{v}_{X_i}^{-s} \mid \bar{t}_s) JS_{\Pi_{\mathbf{v}_{X_i}^{-s}}}[p(x_i \mid \mathbf{v}_{X_i}^{-s}, t_s^\ell), p(x_i \mid \mathbf{v}_{X_i}^{-s}, t_s^r)] \ . \tag{D.19}$$

However, from Eq. (D.18) we see that the terms in this sum that correspond to the assignments of $\mathbf{V}_{X_i}^{-s}$ in which $T_j = t_j^\ell, t_j^r, \bar{t}_j$ are always zero (since the corresponding $p(\mathbf{v}_{X_i}^{-s} \mid \bar{t}_s)$ is zero). Therefore, a merger in T_j cannot change $\bar{d}_1$, as required.

Lastly, we should take care of the case $s = j$. In this case, $p(\bar{t}_s, \bar{t}_j) = 0$ means that the merger $\{t_s^\ell, t_s^r\} \Rightarrow \bar{t}_s$ refers to merging different values of T_j then t_j^ℓ, t_j^r. As a result, while calculating $\Delta\mathcal{L}(t_s^\ell, t_s^r)$ the assignments of T_j are always different from t_j^ℓ, t_j^r (or $\bar{t}_j$). Thus, again, the merger $\{t_j^\ell, t_j^r\} \Rightarrow \bar{t}_j$ does not affect $\Delta\mathcal{L}(t_s^\ell, t_s^r)$. ▮