ARTICLE **Communicated by Liam Paninski**

Multivariate Information Bottleneck

Noam Slonim
nslonim@princeton.edu
Department of Physics and the Lewis–Sigler Institute for Integrative Genomics, Princeton University, Princeton, NJ 08544, U.S.A.

Nir Friedman
nir@cs.huji.ac.il
School of Computer Science and Engineering, Hebrew University, Jerusalem 91904, Israel

Naftali Tishby
tishby@cs.huji.ac.il
School of Computer Science and Engineering, and Interdisciplinary Center for Neural Computation, Hebrew University, Jerusalem 91904, Israel

The information bottleneck (IB) method is an unsupervised model independent data organization technique. Given a joint distribution, $p(X, Y)$, this method constructs a new variable, T, that extracts partitions, or clusters, over the values of X that are informative about Y. Algorithms that are motivated by the IB method have already been applied to text classification, gene expression, neural code, and spectral analysis. Here, we introduce a general principled framework for multivariate extensions of the IB method. This allows us to consider multiple systems of data partitions that are interrelated. Our approach utilizes Bayesian networks for specifying the systems of clusters and which information terms should be maintained. We show that this construction provides insights about bottleneck variations and enables us to characterize the solutions of these variations. We also present four different algorithmic approaches that allow us to construct solutions in practice and apply them to several real-world problems.

1 Introduction

The volume of available data in a variety of domains has grown rapidly over the past few years. Examples include the consistent growth in the amount of

Preliminary and partial versions of this work appeared in *Proc. of the 17th conf. on Uncertainty in artificial Intelligence (UAI-17)*, 2001, and in *Proc. of Neural Information Processing Systems* (NIPS-14), 2002.

online text and the dramatic increase in the available genomic information. As a result, there is a crucial need for data analysis methods. A major goal in this context is the development of unsupervised data representation methods to reveal the inherent hidden structure in a given body of data. Such methods include various dimension-reduction, geometric embedding, and statistical modeling approaches. Arguably the most fundamental class of such methods are clustering techniques.

At first, the clustering problem seems intuitively clear: similar elements should be assigned to the same cluster and dissimilar ones to different clusters. However, formalizing this notion in a well-defined way is not obvious. Nonetheless, such a formulation is essential in order to obtain an objective interpretation of the results. Indeed, while numerous clustering methods exist, many of them are driven by an algorithm rather than by a clear optimization principle, making their results hard to interpret.

Clustering methods can be roughly divided into two categories. The first is based on a choice of some distance or distortion measure among the data points, which presumably reflects some background knowledge about the data. Such a measure implicitly represents the important attributes of the data and is as important as the choice of the data representation itself. For most problems, a proper choice of the distance measure can be the main practical difficulty and through which much of the arbitrariness of the results can enter. Another class of methods is based on statistical assumptions on the origin of the data. Such assumptions enable the design of a statistical model, such as a mixture of gaussians, where the model parameters are then estimated based on the given data.

Roughly speaking, these approaches represent different frameworks for thinking about the data. For instance, in the statistical approach, one usually thinks of the given data as a sample taken from an underlying distribution, whereas in the distance-based approach, such an assumption is typically not required. Distance-based methods can be further divided into central and pairwise clustering methods. In the latter, one uses the direct distances between the points to partition the data, using various (often graph-theoretical) algorithms. In central clustering, one is provided with a distance function that can be applied to new points and generate cluster centroids by minimizing the expected distance to the data points. Central clustering is more closely related to the statistical modeling, whereas the results of the pairwise methods are often more difficult to interpret.

Quite separated from this hierarchy of clustering techniques, the information bottleneck (IB) method (Tishby, Pereira, & Bialek, 1999; Slonim, 2002) stems from a rather different—information theoretic—perspective. The basic idea is surprisingly simple. While choosing the distance measure is notoriously difficult, often one can naturally specify another relevant variable that should be predicted by the obtained clusters. This specification determines which relevant underlying structure we desire. An illustrative example is the problem of speech recognition. While it is not obvious what

the proper distance measure is among speech utterances, in many applications one would agree that the corresponding text is the relevant variable here. Moreover, this scheme allows different valid ways to cluster the same data. For example, one could also define the relevant variable as the identity of the speaker rather than the text and obtain an entirely different, yet equally valid, quantization of the signal.

A common data type that calls for this type of analysis is co-occurrence data, such as verbs and direct objects in sentences (Pereira, Tishby, & Lee, 1993), words and documents (Baker & McCallum, 1998; Hofmann, 1999; Slonim & Tishby, 2000), tissues and gene expression patterns (Tishby & Slonim, 2000), or galaxies and spectral components (Slonim, Somerville, Tishby, & Lahav, 2001). In most such cases, there is no obvious "correct" measure of similarity between the data items. Thus, one would like to rely purely on the joint statistics of the co-occurrences and organize the data such that the relevant information among the variables is captured in the best possible way.

1.1 The Contributions of This Work. The main contribution of this article is in providing a general formulation for a multivariate extension of the IB method. This extension allows us to consider cases where the clustering is relevant with respect to several variables and multiple systems of clusters are constructed simultaneously.

For example, in symmetric, or two-sided, clustering, we want to find two systems of clusters 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 (Slonim & Tishby, 2000). In parallel clustering we attempt to build several systems of clusters of one variable in order to capture independent aspects of the information it conveys about another, relevant, variable. A possible example is the analysis of gene expression data, where multiple independent distinctions about tissues are relevant for the expression of genes. Furthermore, it is possible to think of more complicated scenarios, where there are more than two observed variables. For example, given many input variables that represent a visual stimulus, we might want to recover a smaller set of features that are most informative about an output variable that here represents the firing pattern of a neuron. A most general formulation should consider the compression of different subsets of the observed variables, while maximizing the information about other predefined subsets. The multivariate IB principle, as suggested in this work, provides such a principled general formulation.

To address this type of problem within the IB framework, we use the concept of multi-information, a natural extension of the pairwise concept of mutual information. Our approach further utilizes the theory of probabilistic graphical models such as Bayesian networks for specifying the trade-off terms: which variables to compress and which information terms

should be maintained. In particular, we use one Bayesian network, denoted as G_{in}, to specify a set of variables that 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 should be predicted by its parents in the network. We formulate the general principle as a trade-off between the multi-information each network carries, where we want to minimize the information maintained by G_{in} and at the same time maximize the information maintained by G_{out}. We show that as with the original IB principle, it is possible to analytically characterize the form of the optimal solutions to the general multivariate trade-off principle.

The original IB principle yielded practical algorithms that could be applied to a variety of real-world problems. Here, we show that all these algorithms are naturally extended to the new conceptual framework and illustrate their applicability on several real-world data: text processing applications, gene expression data analysis, and protein sequence analysis.

2 The Information Bottleneck Method

In the original IB principle (Tishby et al., 1999), the relevance of one variable, X, with respect to another one, Y, is quantified in terms of the mutual information (Cover and Thomas, 1991),

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}.$$

This functional is symmetric and nonnegative, and it equals zero if and only if the variables are independent. It measures how many bits are needed on average to convey the information X has about Y and vice versa.

The IB method is thus based on the availability of two variables, with their (assumed given) joint distribution, $p(X,Y)$, where X is the variable we want to compress and Y is the variable we would like to predict. Using a correspondence between this task and the core problems of Shannon's (1948) information theory, Tishby et al. (1999) formulated it as a trade-off between two types of information terms. The idea is to seek partitions of X values that preserve as much mutual information as possible about Y while losing as much information as possible about the original representation, X. Thus, among all the distinctions made by X, one tries to maintain only those that are most relevant to predict Y. Finding optimal representations is posed as a construction of an auxiliary variable, T, that represents soft partitions of X values through $q(T \mid X)$, such that $I(T;X)$ is minimized while $I(T;Y)$ is maximized. Notice that throughout this article, we denote by $q(\cdot)$ the unknown distributions that involve the representation parameters, T, and by $p(\cdot)$ the distributions that are given as input and do not change during the analysis.

Since T is a compressed representation of X, its distribution should be completely determined given X alone; that is, $q(T \mid X, Y) = q(T \mid X)$, or

$$q(X, Y, T) = p(X, Y)q(T \mid X). \tag{2.1}$$

An equivalent formulation is to require the following Markovian independence relation, known as the IB Markovian relation:

$$T \leftrightarrow X \leftrightarrow Y. \tag{2.2}$$

Tishby et al. (1999) formulated this optimization problem as the minimization of the following IB functional,

$$\mathcal{L}[q(T \mid X)] = I(T; X) - \beta I(T; Y), \tag{2.3}$$

where β is a positive Lagrange multiplier and the minimization takes place over all the normalized $q(T \mid X)$ distributions. The distributions $q(T)$ and $q(Y \mid T)$ that are further involved in this functional must satisfy the probabilistic consistency conditions,

$$\begin{cases} q(t) = \sum_x q(x, t) = \sum_x p(x)q(t \mid x) \\ q(y \mid t) = \frac{1}{q(t)} \sum_x q(x, y, t) = \frac{1}{q(t)} \sum_x p(x, y)q(t \mid x), \end{cases} \tag{2.4}$$

where the IB Markovian relation is used in the last equality.

As shown in Tishby et al. (1999), every stationary point of the IB functional must satisfy

$$q(t \mid x) = \frac{q(t)}{Z(x, \beta)} \exp\left(-\beta D_{KL}[p(y \mid x) \| q(y \mid t)]\right), \tag{2.5}$$

where $D_{KL}[p\|q] = E_p[\log \frac{p}{q}]$ is the familiar Kullback-Leibler (KL) divergence (Cover & Thomas, 1991) and $Z(x, \beta)$ is a normalization (partition) function.

The three equations in equations 2.4 and 2.5 must be satisfied self-consistently at all the stationary points of $\mathcal{L}$. The form of the optimal solution in equation 2.5 is reminiscent of the general form of the rate distortion optimal solution (Cover & Thomas, 1991). However, the effective IB distortion, $d_{IB}(X, T) = D_{KL}[p(Y \mid X) \| q(Y \mid T)]$, emerges here from the trade-off principle rather than being chosen arbitrarily.

The self-consistent condition in equation 2.5, together with the marginalization constraints in equation 2.4, can be turned into an iterative algorithm, similar to expectation maximization (EM) and the Blahut-Arimoto algorithm in information theory (Cover & Thomas, 1991). In particular, finding $\{q(T \mid X),\ q(T),\ q(Y \mid T)\}$ that correspond to a (local) stationary point of $\mathcal{L}$,

involves the following iterations:

$$q^{(m)}(t) = \sum_x p(x) q^{(m)}(t \mid x) \tag{2.6}$$

$$q^{(m)}(y \mid t) = \frac{1}{q^{(m)}(t)} \sum_x p(x, y) q^{(m)}(t \mid x)$$

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

whose general convergence was proved in Tishby et al. (1999).

3 Multivariate Extensions of the IB Method

The original formulation of the IB principle concentrated on the trade-off between the compression of one variable, X, and the information this compression maintains about another, relevant variable, Y. We now describe a more general formulation for a multivariate extension of the IB trade-off principle. The motivation is to provide a similar framework for higher-dimensional data and with different probabilistic dependencies, as found in ample examples of real-world problems. This extension combines the theory of probabilistic graphical models, such as Bayesian networks, and a multivariate extension of the mutual information concept, known as multi-information.

3.1 Bayesian Networks and Multi-Information. A Bayesian network over a set of random variables $\mathbf{X} \equiv \{X_1, \ldots, X_n\}$ is a directed acyclic graph (DAG), G, in which vertices are annotated by names of random variables. For each variable X_i, we denote by $\mathbf{Pa}_{X_i}^G$ the set of parents of X_i in G. We say that a distribution $p(\mathbf{X})$ is consistent with G if it can be factored in the form

$$p(X_1, \ldots, X_n) = \prod_i p(X_i \mid \mathbf{Pa}_{X_i}^G),$$

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

The information that the random variables $\{X_1, \ldots, X_n\} \sim p(X_1, \ldots, X_n)$ share about each other is given by (see, e.g., Studeny & Vejnarova, 1998),

$$\begin{aligned} \mathcal{I}[p(\mathbf{X})] = \mathcal{I}(\mathbf{X}) &= D_{KL}[p(X_1, \ldots, X_n) \| p(X_1) \cdots p(X_n)] \\ &= E_{p(X_1, \ldots, X_n)} \left[\log \frac{p(X_1, \ldots, X_n)}{p(X_1) \cdots p(X_n)} \right]. \end{aligned}$$

This multi-information measures the average number of bits that can be gained by a joint compression of all the variables versus independent compression. Like the mutual information, it is nonnegative and equals zero if and only if all the variables are independent.

When the variables have known independence relations, the multi-information can be simplified as follows.

Proposition 1. *Let $\boldsymbol{X} = \{X_1, \ldots, X_n\} \sim p(\boldsymbol{X})$, and let G be a Bayesian network structure over $\boldsymbol{X}$ such that $p \models G$. Then*

$$\mathcal{I}[p(\boldsymbol{X})] = \mathcal{I}(\boldsymbol{X}) = \sum_i I(X_i; \boldsymbol{Pa}_{X_i}^G).$$

That is, the multi-information is the sum of "local" mutual information terms between each variable and its parents. Notice that even if $p(\mathbf{X})$ is not consistent with G, this sum is well defined, but it may capture only part of the real multi-information. Hence, we introduce the following definition.

Definition 1. *Let $\boldsymbol{X} = \{X_1, \ldots, X_n\} \sim p(\boldsymbol{X})$, and let G be a Bayesian network structure over $\boldsymbol{X}$. The multi-information in $p(\boldsymbol{X})$ with respect to G is defined as:*

$$\mathcal{I}^G[p(\boldsymbol{X})] = \sum_i I(X_i; \boldsymbol{Pa}_{X_i}^G), \tag{3.1}$$

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

If $p \models G$, then $\mathcal{I}[p(\mathbf{X})] = \mathcal{I}^G[p(\mathbf{X})]$, but in general, $\mathcal{I}[p(\mathbf{X})] \geq \mathcal{I}^G[p(\mathbf{X})]$ (see below). In this case, we often want to know how close p is to the distribution class that is consistent with G. We define this notion through the M-projection of p on that class:

$$D_{KL}[p\|G] = \min_{q \models G} D_{KL}[p\|q]. \tag{3.2}$$

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

Proposition 2. *Let $p(\boldsymbol{X})$ be a distribution, and let G be a DAG over $\boldsymbol{X}$. Then,*

$$D_{KL}[p\|G] = D_{KL}[p\|q^*], \tag{3.3}$$

where q^* is given by

$$q^*(\boldsymbol{X}) = \Pi_{i=1}^n p(X_i \mid \boldsymbol{Pa}_{X_i}^G). \tag{3.4}$$

Thus, q^* is equivalent to the factorization of p using the conditional independencies implied by G. This proposition extends the Csiszár-Tusnády (1984) lemma that refers to the special case of $n = 2$ (see also Cover & Thomas, 1991, p. 365).

Next, we provide two possible interpretations of the M-projection distance, $D_{KL}[p\|G]$, in terms of the structure of G.

Proposition 3. *Let $\boldsymbol{X} = \{X_1, \ldots, X_n\} \sim p(\boldsymbol{X})$, and let G be a Bayesian network structure over $\boldsymbol{X}$. Assume that the order $X_1, \ldots, X_n$ is consistent with the DAG G (i.e., $\boldsymbol{Pa}^G_{X_i} \subseteq \{X_1, \ldots, X_{i-1}\}$). Then*

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

As an immediate corollary, we have $\mathcal{I}[p(\mathbf{X})] \geq \mathcal{I}^G[p(\mathbf{X})]$, as mentioned earlier. Thus, $D_{KL}[p\|G]$ can be expressed as a sum of local conditional mutual information terms, where each term corresponds to a possible violation of a Markov independence assumption with respect to 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 (Pearl, 1988). 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, $\mathcal{I}[p(\mathbf{X})] = \mathcal{I}(\mathbf{X})$, and the multi-information when p is forced to be consistent with G, $\mathcal{I}^G[p(\mathbf{X})]$, which in particular cannot be larger. Hence, we can think of $D_{KL}[p\|G]$ as the residual information between the variables that is not captured by the dependencies implied by G.

3.2 Multi-Information Bottleneck Principle. Let us consider a set of random variables, $\mathbf{X} = \{X_1, \ldots, X_n\}$, distributed according to $p(\mathbf{X})$. We assume that $p(\mathbf{X})$ is known and forms the input to our analysis, where finite sample issues are discussed in section 7.

Given $p(\mathbf{X})$, we specify a set of partition variables $\mathbf{T} = \{T_1, \ldots, T_k\}$, such that for each subset of $\mathbf{X}$ that we would like to compress, we specify a corresponding subset of the compression variables $\mathbf{T}$. Recall that in the original IB, the IB Markovian relation, $T \leftrightarrow X \leftrightarrow Y$, defines a solution space with all the distributions over $\{X, Y, T\}$ with $q(X, Y, T) = p(X, Y)q(T \mid X)$. Analogously, we define the solution space in our case through a set of IB

Markovian independence relations that imply that each compression variable, $T_j \in \mathbf{T}$, is completely determined given the variables it represents. This is achieved by introducing a DAG, G_{in}, over $\mathbf{X} \cup \mathbf{T}$ where the variables in $\mathbf{T}$ are leaves. 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 independencies implied by G_{in} correspond to the required set of IB Markovian independence relations. In particular, since we require that every T_j is a leaf, every T_j is independent of all the other variables, given the variables it compresses, $\mathbf{Pa}_{T_j}^{G_{in}}$.[1]

The multivariate IB solution space thus consists of all the distributions $q(\mathbf{X}, \mathbf{T}) \models G_{in}$, for which

$$q(\mathbf{X}, \mathbf{T}) = p(\mathbf{X}) \prod_{j=1}^{k} q(T_j \mid \mathbf{Pa}_{T_j}^{G_{in}}), \tag{3.5}$$

where the free parameters correspond to the stochastic mappings $q(T_j \mid \mathbf{Pa}_{T_j}^{G_{in}})$, and the other unknown $q(\cdot)$ distributions are determined by marginalization over $q(\mathbf{X}, \mathbf{T})$ using the Markovian structure of G_{in}. Analogous to the original IB formulation, the information that we would like to minimize is now given by $\mathcal{I}^{G_{in}}$, where $\mathcal{I}^{G_{in}} = \mathcal{I}(\mathbf{X}, \mathbf{T})$ since $q(\mathbf{X}, \mathbf{T}) \models G_{in}$, that is, this is the real multi-information in $q(\mathbf{X}, \mathbf{T})$. Minimizing this quantity attempts to make the $\mathbf{T}$ variables as independent of the $\mathbf{X}$ variables as possible.

Once G_{in} is defined, we need to specify the relevant information that we wish to preserve. We do that by specifying another DAG, G_{out}, which determines what predicts what. Specifically, for every T_j, we define the variables it should preserve information about as its children in G_{out}. Thus, using definition 1, 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

$$\mathcal{L}^{(1)}[q(\mathbf{X}, \mathbf{T})] = \mathcal{I}^{G_{in}}[q(\mathbf{X}, \mathbf{T})] - \beta \mathcal{I}^{G_{out}}[q(\mathbf{X}, \mathbf{T})], \tag{3.6}$$

where the minimization is done subject to the normalization constraints on the partition distributions, and β is the positive Lagrange multiplier controlling the trade-off.[2] Notice that this functional is a direct generalization of the

[1] It is possible to apply a similar derivation where the $\mathbf{T}$ variables are not required to be leaves in G_{in}. This might be useful in various situations where, for example, $T_{j'}$ is used to design a better code for T_j. A relevant example is the relay channel (Cover & Thomas, 1991, Chap. 14).

[2] Since $I^{G_{out}}$ typically consists of several mutual information terms, in principle it is possible to define a separate Lagrange multiplier for each of these terms. This might be

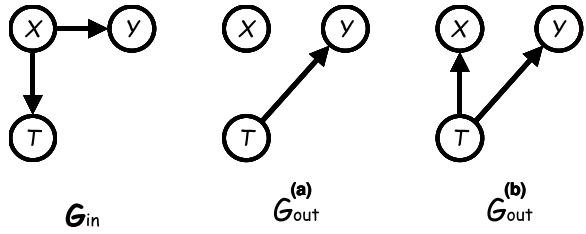

Figure 1: The source (left) and target networks for the original IB principle. The target network for the multivariate IB principle is presented in the middle panel. The target network for the structural principle is described in the right panel.

original IB functional. Again, we are interested in the competition between the complexity of the representation, measured through the compression multi-information, $\mathcal{I}^{G_{in}}$, and the accuracy of the relevant predictions provided by this representation, as quantified by the multi-information $\mathcal{I}^{G_{out}}$.

For $\beta \to 0$, the focus is on the compression term, $\mathcal{I}^{G_{in}}$, yielding a trivial solution in which every T_j consists effectively of one value to which all the values of $\mathbf{Pa}_{T_j}^{G_{in}}$ are mapped, where all the distinctions among these values, relevant or not, are lost. If $\beta \to \infty$, we concentrate on maintaining the relevant information terms as high as possible. This yields a trivial solution at the opposite extreme, where every T_j is a one-to-one map of $\mathbf{Pa}_{T_j}^{G_{in}}$ with no loss of relevant information. The interesting cases are, of course, in between, where β takes positive finite values.

Example 1. Consider the application of the multivariate principle with G_{in} and $G_{out}^{(a)}$ of Figure 1. G_{in} specifies that T compresses X and $G_{out}^{(a)}$ specifies that T should preserve information about Y. For these choices, $\mathcal{I}^{G_{in}} = I(T; X) + I(X; Y)$ and $\mathcal{I}^{G_{out}} = I(T; Y)$. The resulting functional is

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

where $I(X; Y)$ is constant and can be ignored. Thus, we end up with a functional equivalent to that of the original IB functional.

3.3 A Structural Variational Principle. We now describe an alternative and closely related variational principle, which provides more insight

useful if, for example, the preservation of one information term is of greater importance than the preservation of the others.

into the relationship of IB to generative models with maximum-likelihood estimation.

As before, we trade between two complementary goals. On the one hand, we want to compress the observed variables, that is, to minimize $\mathcal{I}^{G_{in}}$. On the other hand, instead of maximizing $\mathcal{I}^{G_{out}}$, we now utilize the compression variables to drive $q(\mathbf{X}, \mathbf{T})$ toward a desired structure, G_{out}, that represents which dependencies and independencies we would like to impose.

Let us consider again the two-variable case shown in Figure 1, where G_{in} specifies that T is a compressed version of X. Ideally, T should preserve all the information about Y. This is equivalent to the situation where T separates X from Y (Cover & Thomas, 1991, Chap. 2), that is, $X \leftrightarrow T \leftrightarrow Y$, as specified by $G^{(b)}_{out}$ of Figure 1. Thus, we wish to construct $q(T \mid X)$ such that the specified independencies in G_{out} are satisfied as much as possible. Notice that in general, G_{in} and G_{out} are incompatible. Except for trivial cases, we cannot achieve both sets of independencies simultaneously (Pereira et al. 1993; Slonim & Weiss, 2002). 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 $q(\mathbf{X}, \mathbf{T})$ be closely approximated by its closest distribution among all distributions consistent with G_{out}. As previously discussed, a natural information theoretic measure of this discrepancy is $D_{KL}[q \| G]$. Thus, the functional that we want to minimize is

$$\mathcal{L}^{(2)}[q(\mathbf{X}, \mathbf{T})] = \mathcal{I}^{G_{in}}[q(\mathbf{X}, \mathbf{T})] + \gamma D_{KL}[q(\mathbf{X}, \mathbf{T}) \| G_{out}], \tag{3.7}$$

where γ is, again, a positive Lagrange multiplier. We will refer to this functional as the structural multivariate IB functional.

The range of γ is between 0, in which case we have the trivial—maximally compressed—solution, and ∞, in which we strive to make q as consistent as possible with G_{out}. Notice that the $\gamma \to \infty$ limit is different in nature from the $\beta \to \infty$ limit, since forcing the dependency structure is different from preserving all the information, as we see next.

Example 2. Consider again the example of Figure 1 with G_{in} and $G^{(b)}_{out}$. In this case, we have $\mathcal{I}^{G_{in}} = I(X; Y) + I(T; X)$ and $\mathcal{I}^{G_{out}} = I(T; X) + I(T; Y)$. From $D_{KL}[q \| G_{out}] = \mathcal{I}^{G_{in}} - \mathcal{I}^{G_{out}}$ (see propostion 3) we obtain

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

where the last (constant) term can be ignored. Hence, we end up with the original IB functional. Thus, we can think of the original IB problem as finding a compression T of X that results in a joint distribution, $q(X, Y, T)$, that is as close as possible to the DAG where X and Y are independent given T.

From proposition 3 we obtain

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

which is similar to the functional $\mathcal{L}^{(1)}$ under the transformation $\beta = \frac{\gamma}{1+\gamma}$. In this transformation, the range $\gamma \in [0, \infty)$ corresponds to the range $\beta \in [0, 1)$. Notice that when $\beta = 1$, we have $\mathcal{L}^{(1)} = D_{KL}[q \| G_{out}]$, 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 $\beta \leq 1$.

As we saw, the two principles require different versions of G_{out} to reconstruct the original IB functional. More generally, for a given principle, different choices of G_{out} yield different optimization problems. Alternatively, given G_{out}, the two principles yield different optimization problems. In the previous example, we saw that these two effects can compensate for each other. In other words, using the structural 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 relation 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 $\mathcal{I}^{G_{in}}$. In the other limit, the two principles differ. When $\beta \to \infty$, in $\mathcal{L}^{(1)}$ we simply maximize $\mathcal{I}^{G_{out}}$. Here, applying $\mathcal{L}^{(1)}$ with $G^{(a)}_{out}$ corresponds to maximizing $I(T; Y)$. However, applying $\mathcal{L}^{(1)}$ with $G^{(b)}_{out}$ corresponds to maximizing $I(T; X) + I(T; Y)$; thus, information about X will be preserved even if it is irrelevant to Y.

When $\gamma \to \infty$, in $\mathcal{L}^{(2)}$ we simply minimize $D_{KL}[q \| G_{out}]$, that is, minimize the violations of conditional independencies implied by G_{out} (see proposition 3). For $G^{(b)}_{out}$, this minimizes $I(X; Y \mid T) = I(X; Y) - I(T; Y)$ (where we used the structure of G_{in} and proposition 3), hence, this is equivalent to maximizing $I(T; Y)$. For $G^{(a)}_{out}$, when $\gamma \to \infty$, we minimize $I(X; Y \mid T) = I(X; Y) + I(T; X) - I(T; Y)$. Thus, unlike the application of $\mathcal{L}^{(1)}$ to $G^{(a)}_{out}$, we cannot ignore the term $I(T; X)$.

To summarize, we can 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 missing or, more precisely, on the conditional independencies implied by their absence. Thus, although both principles can be applied to any choice of G_{out}, some choices make more sense for $\mathcal{L}^{(1)}$ than for $\mathcal{L}^{(2)}$, and vice versa.

3.4 Examples: IB Variations

3.4.1 Parallel IB. In Figure 2A we consider a simple extension of the original IB, where we introduce k compression variables, $\{T_1, \ldots, T_k\}$, instead of one. Similar to the original IB problem, we want $\{T_1, \ldots, T_k\}$ to preserve the information X maintains about Y, as specified by the DAG

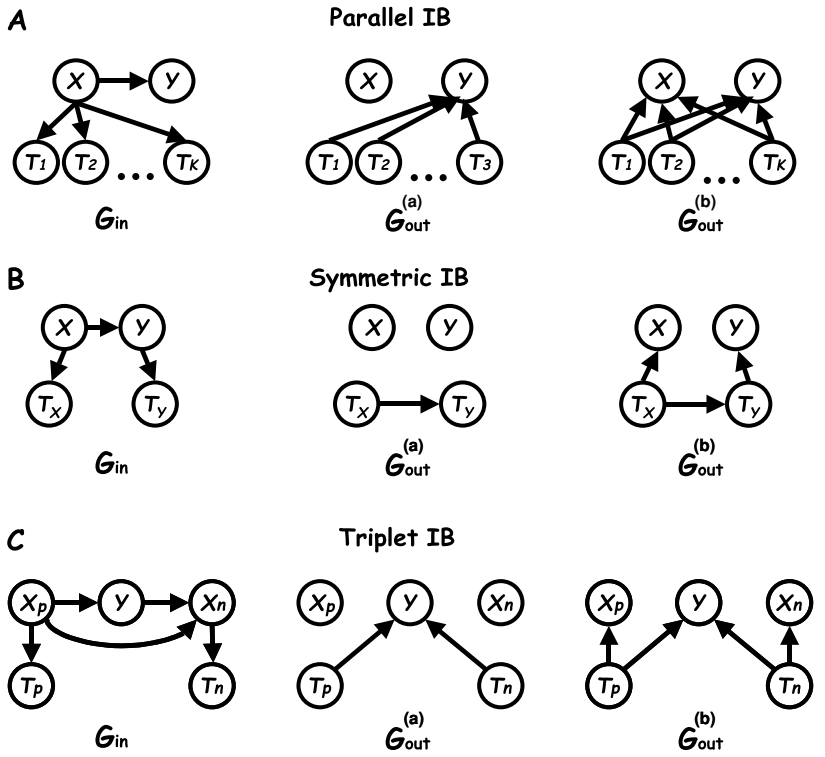

Figure 2: Possible source and target networks for the parallel, symmetric, and triplet IB examples.

$G_{out}^{(a)}$ in the same panel. We call this example the parallel IB, as $\{T_1, \ldots, T_k\}$ compress X in parallel.

Here, $\mathcal{I}^{G_{in}} = I(X; Y) + \sum_{j=1}^{k} I(T_j; X)$ and $\mathcal{I}^{G_{out}^{(a)}} = I(T_1, \ldots, T_k; Y)$; thus,

$$\mathcal{L}_a^{(1)} = \sum_{j=1}^{k} I(T_j; X) - \beta I(T_1, \ldots, T_k; Y). \tag{3.8}$$

That is, 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. From the structure of G_{in}, we can also obtain

$$\sum_{j=1}^{k} I(T_j; X) = I(T_1, \ldots, T_k; X) + \mathcal{I}(T_1, \ldots, T_k), \tag{3.9}$$

where $\mathcal{I}(T_1, \ldots, T_k)$ is the multi-information of all the compression variables. Thus,

$$\mathcal{L}_a^{(1)} = I(T_1, \ldots, T_k; X) + \mathcal{I}(T_1, \ldots, T_k) - \beta I(T_1, \ldots, T_k; Y). \tag{3.10}$$

In other words, we aim to find $\{T_1, \ldots, T_k\}$ that compress X, preserve the information about Y, and remain independent of each other as much as possible.

Recall that using $\mathcal{L}^{(2)}$, we aim at minimizing violation of independencies in G_{out}. This suggests that the DAG $G_{out}^{(b)}$ of Figure 2A captures our intuitions above. In this DAG, X and Y are independent given every T_j and all the T_j's are independent of each other. Here, $\mathcal{I}^{G_{out}^{(b)}} = I(T_1, \ldots, T_k; X) + I(T_1, \ldots, T_k; Y)$, and using equation 3.9, we have

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

which is reminiscent of equation 3.10.

3.4.2 Symmetric IB. Another natural extension of the original IB is the symmetric IB. Here, we want to compress X into T_X and Y into T_Y such that T_X extracts the information X contains about Y while T_Y extracts the information Y contains about X. The DAG G_{in} of Figure 2B captures the form of the compression. For $G_{out}^{(a)}$ in the same panel, we have

$$\mathcal{L}_a^{(1)} = I(T_X; X) + I(T_Y; Y) - \beta I(T_X; T_Y). \tag{3.11}$$

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. Notice that if T_X is informative about T_Y, then it is also informative about Y.

Second, we use the structural principle, $\mathcal{L}^{(2)}$, for which we are interested in approximating the conditional independencies implied by G_{out}. This suggests that $G_{out}^{(b)}$ of Figure 2B represents our desired target model. Here, both T_X and T_Y are sufficient to separate X from Y, while being dependent on each other. Thus, we obtain

$$\mathcal{L}_a^{(2)} = I(T_X; X) + I(T_Y; Y) - \gamma I(T_X; T_Y). \tag{3.12}$$

As in example 1, we see that by using the structural variational principle with a different G_{out}, we end up with the same optimization problem as by using $\mathcal{L}^{(1)}$. Other alternative specifications of G_{out} that are interesting in this

context (Friedman, Mosenzon, Slonim, & Tishby, 2001) are omitted here for brevity.

3.4.3 Triplet IB. A challenging task in the analysis of sequence data, such as DNA and protein sequences or natural language text, is to identify features that are relevant for predicting another symbol in the sequence. Typically these features are different for forward prediction versus backward prediction. For example, the textual features that predict the next word to be *information* are clearly different from those that predict the previous word to be *information*. Here, we address this issue by extracting features of both types such that their combination is highly informative about a symbol between other known symbols.

The DAG G_{in} of Figure 2C 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. Here, 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 that X_p and X_n hold about the current symbol, Y. This is specified by $G^{(a)}_{out}$ in the same panel for which we obtain

$$\mathcal{L}^{(1)}_a = I(T_p; X_p) + I(T_n; X_n) - \beta I(T_p, T_n; Y). \tag{3.13}$$

Second, we use the structural principle, $\mathcal{L}^{(2)}$, with $G^{(b)}_{out}$ of Figure 2C. Here, T_p and T_n are independent, and both are needed to make Y independent of X_p and X_n. Hence, the resulting T_p and T_n partitions provide compact, independent, and informative evidence regarding the value of Y. This specification yields

$$\mathcal{L}^{(2)}_b = I(T_p; X_p) + I(T_n; X_n) - \gamma I(T_p, T_n; Y), \tag{3.14}$$

which is equivalent to equation 3.13. We will term this example the triplet IB.

4 Characterization of the Solution

As shown in Tishby et al. (1999), it is possible to implicitly characterize the form of the optimal solutions to the original IB functional. Here, we provide a similar characterization to the multivariate IB case. Specifically, we want to describe the distributions $q(T_j \mid \mathbf{Pa}^{G_{in}}_{T_j})$ that optimize the trade-off defined by each of the two principles. We present this characterization for $\mathcal{L}^{(1)}$. A similar analysis for $\mathcal{L}^{(2)}$ is straightforward.

We first need some additional notational shorthands. We denote by $\mathbf{U}_j = \mathbf{Pa}^{G_{in}}_{T_j}$ the ($\mathbf{X}$) variables that T_j should compress, by $\mathbf{V}_{X_i} = \mathbf{Pa}^{G_{out}}_{X_i}$ the

variables that should maintain information about X_i, and by $\mathbf{V}_{T_j} = \mathbf{Pa}_{T_j}^{G_{out}}$ the variables that should maintain information about T_j. We also denote $\mathbf{V}_{X_i}^{-j} = \mathbf{V}_{X_i} \setminus \{T_j\}$, and similarly, $\mathbf{V}_{T_\ell}^{-j} = \mathbf{V}_{T_\ell} \setminus \{T_j\}$. To simplify the presentation, we also assume that $\mathbf{U}_j \cap \mathbf{V}_{T_j} = \emptyset$.[3] In addition, we use the notation

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

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

The main result of this section is as follows.

Theorem 1. *Assume that $p(\boldsymbol{X})$, G_{in}, G_{out}, and β are given and that $q(\boldsymbol{X}, \boldsymbol{T}) \models G_{in}$. The conditional distributions $\{q(T_j \mid \boldsymbol{U}_j)\}_{j=1}^{k}$ are a stationary point of $\mathcal{L}^{(1)}[q(\boldsymbol{X}, \boldsymbol{T})] = \mathcal{I}^{G_{in}}[q(\boldsymbol{X}, \boldsymbol{T})] - \beta \mathcal{I}^{G_{out}}[q(\boldsymbol{X}, \boldsymbol{T})]$ if and only if*

$$q(t_j \mid \boldsymbol{u}_j) = \frac{q(t_j)}{Z_{T_j}(\boldsymbol{u}_j, \beta)} e^{-\beta d(t_j, \boldsymbol{u}_j)}, \tag{4.1}$$

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

$$\begin{aligned} d(t_j, \boldsymbol{u}_j) \equiv &\sum_{i: T_j \in \boldsymbol{V}_{X_i}} E_{q(\cdot|\boldsymbol{u}_j)}[D_{KL}[q(X_i \mid \boldsymbol{V}_{X_i}^{-j}, \boldsymbol{u}_j) \| q(X_i \mid \boldsymbol{V}_{X_i}^{-j}, t_j)]] \\ &+ \sum_{\ell: T_j \in \boldsymbol{V}_{T_\ell}} E_{q(\cdot|\boldsymbol{u}_j)}[D_{KL}[q(T_\ell \mid \boldsymbol{V}_{T_\ell}^{-j}, \boldsymbol{u}_j) \| q(T_\ell \mid \boldsymbol{V}_{T_\ell}^{-j}, t_j)]] \\ &+ D_{KL}[q(\boldsymbol{V}_{T_j} \mid \boldsymbol{u}_j) \| q(\boldsymbol{V}_{T_j} \mid t_j)]. \end{aligned} \tag{4.2}$$

The first sum is over all X_i such that T_j participates in predicting X_i. The second sum is over all T_l such that T_j participates in predicting T_l. The last term is related to cases where T_j should be predicted by some $\mathbf{V}_{T_j} \neq \emptyset$.

This theorem provides an implicit set of equations for $q(T_j \mid \mathbf{U}_j)$ through the multivariate relevant distortion $d(T_j, \mathbf{U}_j)$, which in turn depends on

[3] This is, in fact, the standard situation, since $\mathbf{U}_j \cap \mathbf{T} = \emptyset$ and typically $\mathbf{V}_{T_j} \subset \mathbf{T}$.

those unknown distributions. This distortion measures the degree of proximity of the conditional distributions in which $\mathbf{U}_j$ is involved to those where we replace $\mathbf{U}_j$ with its compact representative, T_j. For example, if some cluster $t_j \in \mathcal{T}_j$ behaves more similarly to $\mathbf{u}_j \in \boldsymbol{\mathcal{U}}_j$ than another cluster, $t'_j \in \mathcal{T}_j$, we have $d(t_j, \mathbf{u}_j) < d(t'_j, \mathbf{u}_j)$, which implies $q(t_j \mid \mathbf{u}_j) > q(t'_j \mid \mathbf{u}_j)$. In other words, if t_j is a good representative of $\mathbf{u}_j$ the corresponding membership probability, $q(t_j \mid \mathbf{u}_j)$, is increased accordingly.

As in the original IB problem, equation 4.1 must be solved self-consistently with the equations for the other distributions that involve T_j, which emerge through marginalization over $q(\mathbf{X}, \mathbf{T})$ using the conditional independencies implied by G_{in}.

Notice that when β is small, the $q(T_j \mid \mathbf{U}_j)$ are diffused since β reduces the differences between the distortions for different values of T_j. When $\beta \to \infty$, all the probability mass will be assigned to the value t_j with the smallest distortion, that is, the above stochastic mapping will become deterministic.

4.1 Examples. For G_{in} and $G^{(a)}_{out}$ of Figure 1, it is easy to verify that equation 4.2 amounts to $d(T, X) = D_{KL}[p(Y \mid X) \| q(Y \mid T)]$, in full analogy to equation 2.5, as required.

For the parallel IB case of $G^{(a)}_{out}$ in Figure 2A, we have

$$d(T_j, X) = E_{q(\cdot|X)}[D_{KL}[q(Y \mid \mathbf{T}^{-j}, X) \| q(Y \mid \mathbf{T}^{-j}, T_j)]], \tag{4.3}$$

where we used the notation $\mathbf{T}^{-j} = \mathbf{T} \setminus \{T_j\}$. Notice that due to the structure of G_{in}, $q(Y \mid \mathbf{T}^{-j}, X) = p(Y \mid X)$.

For the symmetric IB case of $G^{(a)}_{out}$ in Figure 2B, we obtain

$$d(T_X, X) = E_{p(\cdot|X)}[D_{KL}[q(T_Y \mid X) \| q(T_Y \mid T_X)]] \tag{4.4}$$

and a symmetric expression for $d(T_Y, Y)$. Thus, T_X attempts to make predictions similar to those of X about T_Y.

Last, for the triplet IB case of $G^{(a)}_{out}$ in Figure 2C, we have

$$d(T_p, X_p) = E_{q(\cdot|X_p)}[D_{KL}[q(Y \mid T_n, X_p) \| q(Y \mid T_n, T_p)]]. \tag{4.5}$$

Thus, $q(t_p \mid x_p)$ 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.

5 Multivariate IB Algorithms

Similar to the original IB functional, the multivariate IB functional is not convex with respect to all of its arguments simultaneously. Except for trivial cases, it always has multiple minima. Since theorem 1 provides necessary

conditions for internal minima of the functional, they can be used to find such solutions. However, as in many other optimization problems, different heuristics can also be employed to construct solutions, with relative advantages and disadvantages. Here, we show that the four algorithmic approaches proposed for the original IB problem (Slonim, 2002) can be extended into the multivariate case. We concentrate on the variational principle $\mathcal{L}^{(1)}$. Deriving the same algorithms for $\mathcal{L}^{(2)}$ is straightforward.

5.1 Iterative Optimization Algorithm: Multivariate iIB. We start with the case where β is fixed. Following Tishby et al. (1999), we apply the fixed-point equations in equation 4.1, alternately with the equations for the other distributions that involve some $\mathbf{T}$ variables. Given the intermediate solution of the algorithm at the mth iteration, $\{q^{(m)}(T_{j'} \mid \mathbf{U}_{j'})\}_{j'=1}^{k}$, we find $q^{(m)}(T_j)$ and $d^{(m)}(T_j, \mathbf{U}_j)$ out of

$$q^{(m)}(\mathbf{X}, \mathbf{T}) = p(\mathbf{X}) \prod_{j'=1}^{k} q^{(m)}(T_{j'} \mid \mathbf{U}_{j'}) \tag{5.1}$$

and then update

$$\begin{cases} q^{(m+1)}(t_j \mid \mathbf{u}_j) \leftarrow \frac{q^{(m)}(t_j)}{Z_{T_j}^{(m+1)}(\mathbf{u}_j, \beta)} e^{-\beta d^{(m)}(t_j, \mathbf{u}_j)}, \\ q^{(m+1)}(t_{j'} \mid \mathbf{u}_{j'}) \leftarrow q^{(m)}(t_{j'} \mid \mathbf{u}_{j'}), \forall j' \neq j. \end{cases} \tag{5.2}$$

In Figure 3 we present pseudocode for this iterative algorithm, which we will term multivariate iIB.

As an example, consider the case of the symmetric IB. Given $\{q^{(m)}(T_X \mid X), q^{(m)}(T_Y \mid Y)\}$, we find $q^{(m)}(T_X)$, $q^{(m)}(T_Y \mid X)$ and $q^{(m)}(T_Y \mid T_X)$ out of $q^{(m)}(X, Y, T_X, Y_Y) = p(X, Y) q^{(m)}(T_X \mid X) q^{(m)}(T_Y \mid Y)$, from which we obtain $d^{(m)}(T_X, X) = D_{KL}[q^{(m)}(T_Y \mid X) \| q^{(m)}(T_Y \mid T_X)]$. Next, we update

$$\begin{cases} q^{(m+1)}(t_x \mid x) \leftarrow \frac{q^{(m)}(t_x)}{Z^{(m+1)}(x, \beta)} e^{-\beta d^{(m)}(t_x, x)}, \\ q^{(m+1)}(t_y \mid y) \leftarrow q^{(m)}(t_y \mid y). \end{cases} \tag{5.3}$$

In the next iteration, we find a new version for $q(T_Y \mid Y)$ while $q(T_X \mid X)$ is kept still. We repeat these updates until convergence to a stationary point. We note that proving convergence in general, for any choice of β and any choice of G_{in} and G_{out}, is more involved than for the original IB problem due to the complex structure of equation 4.2. Nonetheless, in all our experiments, on real and synthetic data, the algorithm always converged to a (locally) optimal solution.

Input: $p(\mathbf{X})$, G_{in}, G_{out}, β, cardinality parameters M_j, convergence parameter ε .

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

Initialization: $m = 0$, randomly initialize $\{q^{(m)}(T_j \mid \mathbf{U}_j)\}_{j=1}^{k}$.

While True

For $j = 1 : k$,

- $q^{(m)}(t_j) \leftarrow \sum_{\mathbf{u}_j} q^{(m)}(t_j \mid \mathbf{u}_j) p(\mathbf{u}_j) \;\; \forall\, t_j \in \mathcal{T}_j$
- Find the mth update for all the distributions in $d(T_j, \mathbf{U}_j)$

  by marginalization w.r.t. $q^{(m)}(\mathbf{X}, \mathbf{T}) = p(\mathbf{X}) \prod_{j'=1}^{k} q^{(m)}(T_{j'} \mid \mathbf{U}_{j'})$
- $q^{(m+1)}(t_j \mid \mathbf{u}_j) \leftarrow \frac{q^{(m)}(t_j)}{Z_{T_j}^{(m+1)}(\mathbf{u}_j, \beta)} e^{-\beta d^{(m)}(t_j, \mathbf{u}_j)} \;\; \forall\, t_j \in \mathcal{T}_j,\; \forall\, \mathbf{u}_j \in \mathcal{U}_j$
- $q^{(m+1)}(T_l \mid \mathbf{U}_l) \leftarrow q^{(m)}(T_l \mid \mathbf{U}_l) \;\; \forall\, l = 1 : k,\; l \neq j$
- $m \leftarrow m + 1$

If $\forall\, j = 1 : k,\; \forall\, \mathbf{u}_j \in \mathcal{U}_j, \;\; JS_{\frac{1}{2},\frac{1}{2}}[q^{(m+1)}(T_j \mid \mathbf{u}_j), q^{(m)}(T_j \mid \mathbf{u}_j)] \leq \varepsilon$,

break

Figure 3: Pseudocode of the multivariate iterative IB algorithm (multivariate iIB). *JS* denotes the Jensen-Shannon divergence (see equation 5.8). In principle, we repeat this procedure for different initializations and choose the solution that minimizes $\mathcal{L} = \mathcal{I}^{G_{in}} - \beta \mathcal{I}^{G_{out}}$.

5.2 Deterministic Annealing Algorithm: Multivariate dIB. It is often desirable to explore a hierarchy of solutions at different β values. Thus, we now present a multivariate deterministic annealing-like procedure that extends the original approach in Tishby et al. (1999).

In deterministic annealing, we iteratively increase β and then adapt the solution at the previous β value to the new one (Rose, 1998). Recall that for $\beta \to 0$, the solution consists of essentially one cluster per T_j. As β is increased, at some critical point the values of some T_j diverge and show different behaviors. Successive increments of β will reach additional bifurcations that we wish to identify. Thus, for each T_j, we end up with a bifurcating hierarchy that traces the sequence of solutions at different β values.

To detect these bifurcations, we adopt the method suggested in Tishby et al. (1999). Given the solution from the previous β value, we construct an initial problem in which every T_j value is duplicated. Let t_j^{ℓ} and t_j^{r} be

Input: $p(\mathbf{X})$, G_{in}, G_{out}, β, M_j, α, ε_β, and $b^{(j)}$.

Output: "Soft" partitions T_j of $\boldsymbol{\mathcal{U}}_j$ into $m = 1, \ldots, M_j$ clusters $\forall j = 1 : k$.

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

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:

$$q^*(t_j^\ell \mid \mathbf{u}_j) = q(t_j \mid \mathbf{u}_j)\left(\tfrac{1}{2} + \alpha\hat{\epsilon}(t_j, \mathbf{u}_j)\right),\ q^*(t_j^r \mid \mathbf{u}_j) = q(t_j \mid \mathbf{u}_j)\left(\tfrac{1}{2} - \alpha\hat{\epsilon}(t_j, \mathbf{u}_j)\right)$$

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 $JS_{\frac{1}{2},\frac{1}{2}}[q(\mathbf{Ne}_{T_j}^{G_{out}} \mid t_j^\ell), q(\mathbf{Ne}_{T_j}^{G_{out}} \mid t_j^r)] \geq b^{(j)}$,

$\mathcal{T}_j \leftarrow \{\mathcal{T}_j \setminus \{t_j\}\} \cup \{t_j^\ell, t_j^r\}$

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

break

Figure 4: Pseudocode of the multivariate deterministic annealing-like algorithm (multivariate dIB). *JS* denotes the Jensen-Shannon divergence (see equation 5.8). $\mathbf{Ne}_{T_j}^{G_{our}}$ denotes the neighbors of T_j in G_{out} (parents/direct descendants). $f(\beta, \varepsilon_\beta)$ is a simple function used to increase β based on its current value and on some scaling parameter ε_β.

two such duplications of $t_j \in \mathcal{T}_j$. Then we set $q^*(t_j^\ell \mid \mathbf{u}_j) = q(t_j \mid \mathbf{u}_j)\left(\frac{1}{2} + \alpha\hat{\epsilon}(t_j, \mathbf{u}_j)\right)$ and $q^*(t_j^r \mid \mathbf{u}_j) = q(t_j \mid \mathbf{u}_j)\left(\frac{1}{2} - \alpha\hat{\epsilon}(t_j, \mathbf{u}_j)\right)$, where $\hat{\epsilon}(t_j, \mathbf{u}_j)$ is a randomly drawn noise term and $\alpha > 0$ is a (small) scale parameter. Thus, each copy is a slightly perturbed version of t_j. For large enough β, this perturbation suffices for the two copies to diverge; otherwise, they collapse to the same solution.

Specifically, given this initial point, we apply the multivariate iIB algorithm. After convergence, if t_j^ℓ and t_j^r are sufficiently different, we declare that 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 pseudocode is given in Figure 4.

There are several difficulties with this algorithm. The parameters $b^{(j)}$ that are involved in detecting the bifurcations often should scale with β. Further, one may need to tune the rate of increasing β; otherwise, cluster splits might be skipped. Last, the duplication process is stochastic in nature and involves additional parameters. Some of these issues were addressed rigorously for the original IB problem (Parker, Gedeon, & Dimitrov, 2002). Extending this work in our context seems like a natural direction for future research.

5.3 Agglomerative Algorithm: Multivariate aIB. The agglomerative IB algorithm was introduced in Slonim & Tishby (1999) as a simple and approximated algorithm for the original IB problem. It employs a greedy agglomerative clustering technique to find a hierarchical clustering tree in a bottom-up fashion and was found to be useful for various problems (Slonim, 2002). We now present a multivariate extension of this algorithm. To this aim, it is more convenient to consider the problem of maximizing

$$\mathcal{L}_{\max}[q(\mathbf{X}, \mathbf{T})] = \mathcal{I}^{G_{out}}[q(\mathbf{X}, \mathbf{T})] - \beta^{-1} \cdot \mathcal{I}^{G_{in}}[q(\mathbf{X}, \mathbf{T})], \tag{5.4}$$

which is clearly equivalent to minimizing $\mathcal{L}^{(1)}$ (see equation 3.6).

Our algorithm starts with the most fine-grained solution, $T_j = \mathbf{U}_j$. Thus, every $\mathbf{u}_j$ is solely assigned to a unique singleton cluster, $t_j \in \mathcal{T}_j$, and the assignment probabilities, $\{q(T_j \mid \mathbf{U}_j)\}_{j=1}^{k}$ are deterministic—either 0 or 1 (that is, "hard" clustering). Nonetheless, the following analysis holds for the general case of soft clustering as well.

Given the singleton initialization, we reduce the cardinality of one T_j by agglomerating, or merging, two of its values, t_j^{ℓ} and t_j^{r}, into a single value $\bar{t}_j$. Formally, this is defined through

$$q(\bar{t}_j \mid \mathbf{u}_j) = q(t_j^{\ell} \mid \mathbf{u}_j) + q(t_j^{r} \mid \mathbf{u}_j). \tag{5.5}$$

The corresponding conditional merger distribution is defined through

$$\Pi_{\mathbf{z}} = \{\pi_{\ell,\mathbf{z}}, \pi_{r,\mathbf{z}}\} = \left\{ \frac{q(t_j^{\ell} \mid \mathbf{z})}{q(\bar{t}_j \mid \mathbf{z})}, \frac{q(t_j^{r} \mid \mathbf{z})}{q(\bar{t}_j \mid \mathbf{z})} \right\}. \tag{5.6}$$

Note that if $\mathbf{Z} = \emptyset$, then $\Pi_{\mathbf{z}} = \Pi = \{\frac{q(t_j^{\ell})}{q(\bar{t}_j)}, \frac{q(t_j^{r})}{q(\bar{t}_j)}\}$, that is, these are the relative weights of each of the clusters that participate in the merger (Slonim & Tishby, 1999).

The basic question in an agglomerative process is which pair to merge at each step. 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. Then our merger cost is

given by

$$\Delta\mathcal{L}_{\max}(t_j^\ell, t_j^r) = \mathcal{L}_{\max}^{bef} - \mathcal{L}_{\max}^{aft}, \tag{5.7}$$

where $\mathcal{L}_{\max}^{bef}$ and $\mathcal{L}_{\max}^{aft}$ are calculated based on T_j^{bef} and T_j^{aft}, respectively. The greedy procedure evaluates all the potential mergers, for every T_j, and applies the one that minimizes $\Delta\mathcal{L}_{\max}(t_j^\ell, t_j^r)$. This is repeated until all the $\mathbf{T}$ variables degenerate into trivial clusters. The resulting set of hierarchies describes a range of solutions at all the different resolutions.

A direct calculation of all the potential merger costs using equation 5.7 is typically unfeasible. However, as in Slonim & Tishby (1999), 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. An essential concept in this derivation is the Jensen-Shannon (*JS*) divergence. Specifically, the *JS* divergence between two probability distributions, p_1 and p_2, with respect to the positive weights, $\Pi = \{\pi_1, \pi_2\}$, $\pi_1 + \pi_2 = 1$, is given by

$$JS_\Pi[p_1, p_2] = \pi_1 D_{KL}[p_1 \| \bar{p}] + \pi_2 D_{KL}[p_2 \| \bar{p}], \tag{5.8}$$

where $\bar{p} = \pi_1 p_1 + \pi_2 p_2$. The *JS* is nonnegative and upper bounded, and it equals zero if and only if $p_1 = p_2$. It is also symmetric, but it does not satisfy the triangle inequality. Comparing the *JS* between two empirical distributions of two samples with some predefined threshold is asymptotically the optimal way to determine whether both samples came from a single source (Gutman, 1989; Schreibman, 2000).

Theorem 2. *Let $t_j^\ell, t_j^r \in \mathcal{T}_j$ be two clusters. Then,*

$$\Delta\mathcal{L}_{max}(t_j^\ell, t_j^r) = q(\bar{t}_j) \cdot d_A(t_j^\ell, t_j^r), \tag{5.9}$$

where

$$\begin{aligned} d_A(t_j^\ell, t_j^r) \equiv & \sum_{i:T_j \in \boldsymbol{V}_{X_i}} E_{q(\cdot|\bar{t}_j)}[JS_{\Pi_{\boldsymbol{V}_{X_i}^{-j}}}[q(X_i \mid \boldsymbol{V}_{X_i}^{-j}, t_j^\ell), q(X_i \mid \boldsymbol{V}_{X_i}^{-j}, t_j^r)]] \\ & + \sum_{\ell:T_j \in \boldsymbol{V}_{T_\ell}} E_{q(\cdot|\bar{t}_j)}[JS_{\Pi_{\boldsymbol{V}_{T_\ell}^{-j}}}[q(T_\ell \mid \boldsymbol{V}_{T_\ell}^{-j}, t_j^\ell), q(T_\ell \mid \boldsymbol{V}_{T_\ell}^{-j}, t_j^r)]] \\ & + JS_\Pi[q(\boldsymbol{V}_{T_j} \mid t_j^\ell), q(\boldsymbol{V}_{T_j} \mid t_j^r)] \\ & - \beta^{-1} \cdot JS_\Pi[q(\boldsymbol{U}_j \mid t_j^\ell), q(\boldsymbol{U}_j \mid t_j^r)]. \end{aligned} \tag{5.10}$$

That is, the merger cost is a multiplication of the "weight" of the merger components, $q(\bar{t}_j)$, with their "distance," $d_A(t_j^\ell, t_j^r)$. Notice that due to the *JS*

Input: $p(\mathbf{X})$, G_{in}, G_{out}, β .

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

Initialization:

For $j = 1 : k$, $T_j \leftarrow \mathbf{U}_j$.

For $j = 1 : k$,

$\forall\, t_j^\ell, t_j^r \in \mathcal{T}_j$,

calculate $\Delta\mathcal{L}_{\max}(t_j^\ell, t_j^r) = q(\bar{t}_j) \cdot d_A(t_j^\ell, t_j^r)$

Main Loop

While there is j with $|\mathcal{T}_j| > 1$,

$\{j, \ell, r\} = argmin_{j',\ell',r'} \Delta\mathcal{L}_{\max}(t_{j'}^{\ell'}, t_{j'}^{r'})$

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. $\bar{t}_j$

Figure 5: Pseudocode of the multivariate agglomerative IB algorithm (multivariate aIB).

properties, this "distance" is symmetric, but it is not a metric. It is small for pairs that give similar predictions about the variables that T_j should predict, and have different predictions, or minimum overlap about the variables that T_j should compress.

Equation 5.10 is clearly analogous to equation 4.2. While for the multivariate iIB the optimization is governed by the *KL* divergences between data and cluster centroids, here it is controlled through *JS* divergences, which are related to the likelihood that the two merged clusters have a common source. For brevity, in the rest of this section, we focus on the simpler hard clustering case, for which $JS_\Pi[q(\mathbf{U}_j \mid t_j^\ell), q(\mathbf{U}_j \mid t_j^r)] = H[\Pi]$, where H is Shannon's entropy. A pseudocode of the general procedure is given in Figure 5.

Examples. For the original IB problem (G_{in} and $G_{out}^{(a)}$ in Figure 1), we obtain

$$\Delta\mathcal{L}_{\max}(t^l, t^r) = q(\bar{t}) \cdot (JS_\Pi[q(Y \mid t^l), q(Y \mid t^r)] - \beta^{-1} H[\Pi]), \tag{5.11}$$

which is consistent with the original aIB algorithm (Slonim & Tishby, 1999).

For the parallel IB (G_{in} and $G_{out}^{(a)}$ in Figure 2A), we have

$$\Delta\mathcal{L}_{\max}(t_j^\ell, t_j^r) = q(\bar{t}_j) \cdot (E_{q(\cdot|\bar{t}_j)}[JS_{\Pi_{\mathbf{T}^{-j}}}[q(Y \mid \mathbf{T}^{-j}, t_j^\ell), q(Y \mid \mathbf{T}^{-j}, t_j^r)]] - \beta^{-1} H[\Pi]), \tag{5.12}$$

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

For the symmetric IB (G_{in} and $G_{out}^{(a)}$ in Figure 2B), we obtain

$$\Delta\mathcal{L}_{\max}(t_X^\ell, t_X^r) = q(\bar{t}_X) \cdot (JS_\Pi[q(T_Y \mid t_X^\ell), q(T_Y \mid t_X^r)] - \beta^{-1} H[\Pi]), \tag{5.13}$$

and an analogous expression for T_Y.

5.4 Sequential Optimization Algorithm: Multivariate sIB. An agglomerative approach is relatively computationally demanding. If we start from $T_j = \mathbf{U}_j$, the time complexity is $O(\sum_{j=1}^k |\boldsymbol{\mathcal{U}}_j|^3 \mid \boldsymbol{\mathcal{V}}_j|)$ (where $\mid \boldsymbol{\mathcal{V}}_j|$ denotes the complexity of calculating a single merger in T_j), while the space complexity is $O(\sum_{j=1}^k |\boldsymbol{\mathcal{U}}_j|^2)$, that is, unfeasible for large data sets. Moreover, it is not guaranteed to extract even locally optimal solutions. Recently, we suggested a framework for casting an agglomerative clustering algorithm into a sequential optimization algorithm, which is guaranteed to converge to a stable solution in much better time and space complexity (Slonim, Friedman, & Tishby, 2002). Next, we describe how to apply this idea in our context.

The sequential procedure maintains for each T_j a flat partition with M_j (hard) clusters. At each step we draw a $\mathbf{u}_j \in \boldsymbol{\mathcal{U}}_j$ out of its current cluster (denoted here $t_j(\mathbf{u}_j)$) and represent it as a new singleton cluster. Using equation 5.9, we now merge $\mathbf{u}_j$ into a cluster t_j^{new} such that $t_j^{new} = \text{argmin}_{t_j \in \mathcal{T}_j} \Delta\mathcal{L}_{\max}(\{\mathbf{u}_j\}, t_j)$, to obtain a (possibly new) partition T_j^{new}, with the appropriate cardinality. Since this step can only increase the (upper-bounded) functional $\mathcal{L}_{\max}$, we are guaranteed to converge to a stable solution.

It is easy to verify that our time complexity is $O(\ell \cdot \sum_j |\boldsymbol{\mathcal{U}}_j \| \mathcal{T}_j \| \boldsymbol{\mathcal{V}}_j|)$, where ℓ is the number of iterations we should perform until convergence is attained. Since typically $\ell \cdot |\mathcal{T}_j| \ll |\boldsymbol{\mathcal{U}}_j|^2$, we get a significant run-time improvement. Moreover, we dramatically improve our memory consumption toward $O(\sum_j |\mathcal{T}_j|^2)$. We will term this algorithm multivariate sIB. A pseudocode is given in Figure 6.

6 Illustrative Applications

In this section we consider a few illustrative applications of the general methodology. In practice we do not have access to the true joint distribution, $p(\mathbf{X})$, but only a finite sample drawn out of this distribution. Here, a

Input: $p(\mathbf{X})$, G_{in}, G_{out}, β, cardinality parameters M_j .

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

Initialization:

For $j = 1 : k$,

$T_j \leftarrow$ random partition of $\boldsymbol{\mathcal{U}}_j$ into M_j clusters.

While True

$Done \leftarrow TRUE$

For $j = 1 : k$,

$\forall\, \mathbf{u}_j \in \boldsymbol{\mathcal{U}}_j$

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 $t_j^{new}(\mathbf{u}_j)$

If $Done$,

break

Figure 6: Pseudocode of the multivariate sequential IB algorithm (multivariate sIB). In principle, we repeat this procedure for different initializations and choose the solution that maximizes $\mathcal{L}_{\max} = \mathcal{I}^{G_{out}} - \beta^{-1}\mathcal{I}^{G_{in}}$.

pragmatic approach was taken where we estimated $p(\mathbf{X})$ through a simple normalization. Our results seem satisfactory even in extreme undersampling situations, and we leave the theoretical analysis of the finite sample effects over our methodology for future research.

An appealing property of an information-theoretic approach, and the IB framework in particular, is that it can be applied to data sets of different nature in exactly the same manner. There is no need to tailor the algorithms to the given data or to define a domain-specific distortion measure. To demonstrate this issue, we apply our method to a variety of data types, including natural language text, protein sequences, and gene expression data (see appendix B regarding preprocess and implementation details). In all cases, the quality of the results can be assessed on similar grounds in terms of compression versus preservation of relevant information.

6.1 Parallel IB Applications. We consider the specification of G_{in} and $G_{out}^{(a)}$ of Figure 2A. The problem thus is to partition X into k sets of clusters, $\mathbf{T} = \{T_1, \ldots, T_k\}$, that minimize the information they maintain about X, maximize the information they hold about Y, and remain indepenent of each other as much as possible.

For various technical reasons, the sIB algorithm is most suitable here. Specifically, here, we first apply sIB with $k = 1$, which is equivalent to solving the original IB problem. Given this solution, denoted as T_1, we apply again sIB with $k = 2$, while T_1 is kept fixed. That is, given T_1, we look for T_2 such that $I(T_1, T_2; Y)) - \beta^{-1}(I(T_1, T_2; X) + I(T_1; T_2))$ is maximized. Next, we hold T_1 and T_2 fixed while looking 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 a second—approximately independent—principal partition, and so on.

6.1.1 Parallel sIB for Style Versus Topic Text Clustering. A well-known concern in cluster analysis is that there might be more than one meaningful way to partition a given body of data. For example, text documents might have two possible dichotomies: by their topics and by their writing styles. Next, we construct such an example and solve it using our parallel IB approach.

We selected four books: *The Beasts of Tarzan* and *The Gods of Mars* by E. R. Burroughs and *The Jungle Book* and *Rewards and Fairies* by R. Kipling. Thus, except for the partition according to the writing style, there is a possible topic partition of the "jungle" topic versus all the rest. We split each book into "documents" consisting of 200 successive words each. We defined $p(w, d)$ as the number of occurrences of the word w in the document d, normalized by the total number of words in the corpus, and applied parallel sIB to cluster the documents into two partitions, T_1 and T_2, of two clusters each.

Since this setting already implies significant compression, we used $\beta^{-1} = 0$ and concentrated on maximizing $I(T_1, T_2; W)$. In Table 1, we see that T_1 shows almost perfect correlation to an authorship partitioning, while T_2 is correlated with a topical partitioning. Moreover, $I(T_1; T_2) \approx 0.001$ *nats*, that is, these two partitions are practically independent. In addition, with only four clusters, $I(T_1, T_2; W) \approx 0.3$ *nats*, which is about 13% of the original (empirical) information, $I(D; W)$.

6.1.2 Parallel sIB for Gene Expression Data Analysis. As our second example, we used the mRNA gene expression measurements of approximately 6800 human genes in 72 samples of leukemia (Golub et al., 1999). These data include independent annotations of their components, including the type of leukemia (*ALL* versus *AML*), the type of cells, the donating hospital, and others. We normalized the measurements of the genes in each sample independently to get an estimated joint distribution, $p(S, G)$, over samples and genes (where $p(S)$ is uniform). Given this joint distribution, we

Table 1: Results for Parallel sIB Applied to Style Versus Topic Text Clustering.

	$T_{1,a}$	$T_{1,b}$	$T_{2,a}$	$T_{2,b}$
The Beasts 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

Notes: Each entry indicates the number of "documents" in some cluster and some class. For example, the first cluster of the first partition, $T_{1,a}$, includes 315 "documents" taken from the book *The Beast of Tarzan*.

Table 2: Results for Parallel sIB Applied to Gene Expression Measurements of Leukemia Samples (Golub et al., 1999).

	$T_{1,a}$	$T_{1,b}$	$T_{2,a}$	$T_{2,b}$	$T_{3,a}$	$T_{3,b}$	$T_{4,a}$	$T_{4,b}$
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

Note: Each entry indicates the number of samples in some cluster and some class. Note that T-cell/B-cell annotations are available only for samples annotated as ALL type. The last row indicates the average "prediction strength" score (Golub et al., 1999) in the cluster.

applied the parallel sIB to cluster the samples into four clustering hierarchies, consisting of two clusters each (again, with $\beta^{-1} = 0$).

In Table 2 we present the four partitions. T_1 almost perfectly matches the *AML* versus *ALL* annotation, while T_2 is correlated with the B-cells/T-cell split. For T_3 we note that the average "prediction strength" score (see Golub et al., 1999) is very different between both clusters. For T_4 we were not able to find any clear correlation with the available annotations, suggesting that this partition overfits the data or that further meaningful partitions of these data are not expressed in the provided annotations. In terms of information, $I(\mathbf{T}; G)$ preserves about 54% of the original information, $I(S; G) \approx 0.23\,nats$.

6.2 Symmetric IB Applications. Here, we illustrate the applicability of all our algorithms for G_{in} and $G_{out}^{(a)}$ of Figure 2B.

6.2.1 Symmetric dIB and iIB for Word-Topic Clustering. We start with a simple text processing example, constructed out of the 20 news-groups data (Lang, 1995). These data consist of approximately 20,000 documents, distributed among 20 different topics. We defined $p(w, c)$ as the number of occurrences of a word w in all documents of topic c, normalized by

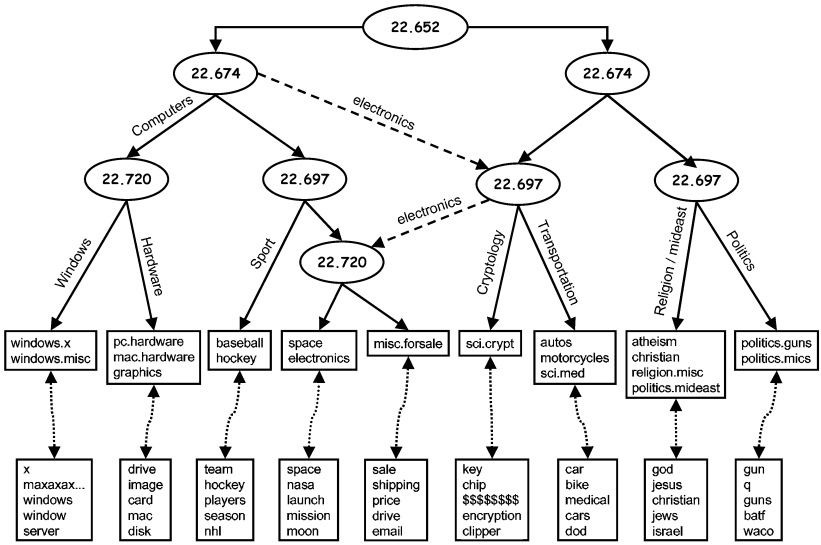

Figure 7: Application of the symmetric dIB to the 20-news-group data. The learned hierarchy of topic clusters, T_c, is presented after four splits. The numerical value inside each ellipse denotes the bifurcation β value. Notice that at the early stages, the algorithm is inconclusive regarding the assignment of the *electronics* topic, which demonstrates that the hierarchy obtained by the dIB algorithm does not necessarily construct a tree. Below every topic cluster, t_c, we present its most probable word cluster ($t_w^* = argmax_{t_w \in \mathcal{T}_w} q(t_w \mid t_c)$) through its five most probable words, sorted via $q(w \mid t_w^*)$.

the total number of words in the corpus, and applied the symmetric dIB algorithm. We start with $\beta^{-1} = 0$ and gradually "anneal" the system to extract a hierarchy of word clusters, T_w, and a corresponding hierarchy of topic clusters, T_c.

The obtained T_c partitions were typically hard; hence, this hierarchy can be presented as a simple tree-like structure (see Figure 7). For every t_c, we find its most probable word cluster ($t_w^* = \mathrm{argmax}_{t_w} q(t_w \mid t_c)$) and present it in the same figure. Evidently there is a strong semantic relation in every such pair.

The word clusters also utilized the soft clustering utility to deal with words that are relevant to several topics, as illustrated in Table 3. In terms of information, after four splits, $|\mathcal{T}_w| = 14$, $|\mathcal{T}_c| = 9$, and $I(T_w; T_c) = 0.6\,nats$, which is about 70% of the original information, $I(W; C)$.

We further applied the symmetric iIB algorithm to the same data. For purposes of comparison, the input parameters were set as in the dIB result, $|\mathcal{T}_w| = 14$, $|\mathcal{T}_c| = 9$, and $\beta \approx 22.72$. We performed 100 different random initializations, 8 of which converged to a better minimum of $\mathcal{L}$ than the one

Table 3: Results for "Soft" Word Clustering Using Symmetric dIB over the 20-News-group Data.

W	$q(t_w \mid w)$	t_c^*	$q(t_c^* \mid t_w)$
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

Notes: The first column indicates the word, that is, the W value. The second column presents $q(t_w \mid w)$ for all the clusters for which this probability was nonzero. t_c^* denotes the topic cluster that maximizes $q(t_c \mid t_w)$. It is represented in the table, in the third column, by the joint topic of its members (see Figure 7). The last column presents the probability of this topic cluster given t_w.

found by dIB (see Figure 8). Thus, by tracking the changes in the solution as β increases, the dIB approach succeeds in finding a relatively good solution and also provides more details by describing a hierarchy of solutions. Nonetheless, if one is interested in a flat partition, applying iIB with a sufficient number of initializations will probably yield a better optimum.

6.2.2 Symmetric sIB and aIB for Protein Sequence Analysis. As a second example we used a subset of five protein classes taken from the PRINTS database (Attwood et al., 2000) (see Table 4). All five classes share a common protein structural unit, known as the glutathione S-transferase (GST) domain. A well-established database (the Pfam database, http://www.sanger.ac.uk/Pfam) has chosen not to model these groups separately due to high sequence similarity between them. Nonetheless, unsupervised symmetric IB algorithms extract clusters that are well correlated with these groups.

We represented each protein as a count vector over different 4-mers of amino acids, or features, present in these data. We defined $p(f \mid r)$ as the relative frequency of a feature f in a protein r and further defined $p(r)$ as uniform. Given these data, we applied the symmetric aIB and sIB (with $\beta^{-1} = 0$) to extract protein clusters, T_R, and feature clusters, T_F, such that $I(T_R; T_F)$ is maximized.

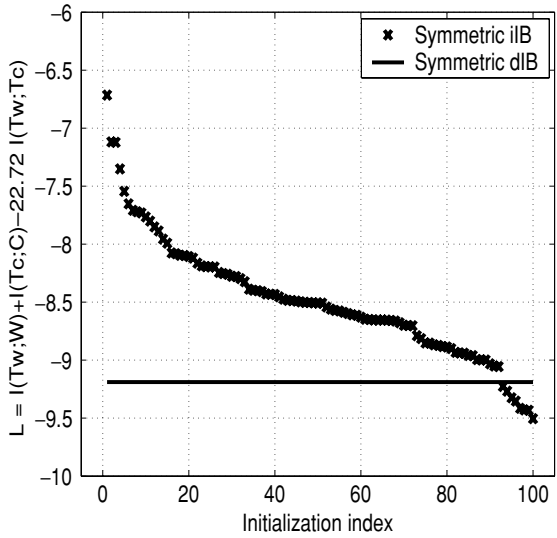

Figure 8: Application of symmetric dIB and symmetric iIB to the 20-news-group data. The iIB results over 100 different initializations are sorted with respect to $\mathcal{L} = I(T_w; W) + I(T_c; C) - 22.72 \cdot I(T_w; T_c)$. In eight cases, *iIB* converged to a better minimum.

Table 4: Data Set Details of the Protein GST Domain Test.

Class	Family Name	Number of Proteins
c_1	*GST*—no class label	298
c_2	*S crystalline*	29
c_3	*Alpha class GST*	40
c_4	*Mu class GST*	32
c_5	*Pi class GST*	22

For the sIB results, with 10 protein clusters and 10 feature clusters, we obtain $I(T_R; T_F) = 1.1$ *nats* ($\sim 30\%$ of the original information), and the algorithm almost perfectly recovers the manual biological partitioning of the proteins (see Table 5).

For each t_R, we identify its most probable feature cluster ($t_F^* = \text{argmax}_{t_F} q(t_F \mid t_R)$) and present in Table 6 its most probable features, which apparently are good indicators for the biological class that is correlated with t_R.

Table 5: Results for Applying Symmetric sIB to the GST Protein Data Set with $|\mathcal{T}_R| = 10$, $|\mathcal{T}_F| = 10$.

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

Notes: Each entry indicates the number of proteins in some cluster and some class. The last row indicates the number of "errors" for each cluster, defined as the number of proteins in this cluster that are not labeled by the cluster's most dominant label.

Table 6: Results for Symmetric sIB: Indicative Features for GST Protein Classes.

T_R	t_F^*	$q(t_F^*\|t_R)$	Feature	$q(f\|t_F^*)$	c_1	c_2	c_3	c_4	c_5
t_{R_7}	$t_{F_{10}}$	0.91	*RYLA*	0.022	0.08	1.00	0.00	0.19	0.00
(c_2)			*GRGR*	0.02	0.04	0.72	0.58	0.00	0.00
			NGRG	0.019	0.04	0.72	0.43	0.00	0.00
t_{R_9}	t_{F_8}	0.89	*FPNL*	0.025	0.06	0.00	0.00	1.00	0.00
(c_4)			*AILR*	0.018	0.03	0.00	0.03	0.88	0.64
			SNAI	0.017	0.03	0.00	0.03	0.94	0.46
t_{R_10}	t_{F_2}	0.85	*LDLL*	0.019	0.04	0.00	0.08	0.00	0.59
(c_5)			*SFAD*	0.017	0.04	0.00	0.03	0.00	0.64
			FETL	0.017	0.04	0.00	0.03	0.00	0.59
t_{R_8}	t_{F_1}	0.85	*FPLL*	0.018	0.01	0.00	0.90	0.00	0.77
(c_3)			*YGKD*	0.017	0.03	0.00	0.88	0.00	0.50
			AAGV	0.016	0.03	0.45	0.78	0.00	0.00
t_{R_5}	t_{F_9}	0.83	*TLVD*	0.015	0.13	0.00	0.00	0.00	0.00
(c_1)			*WESR*	0.015	0.12	0.00	0.00	0.00	0.00
			EFLK	0.015	0.00	0.00	0.00	0.19	0.00
t_{R_4}	t_{F_5}	0.80	*IPVL*	0.010	0.11	0.00	0.00	0.00	0.00
(c_1)			*ARFW*	0.010	0.11	0.00	0.00	0.00	0.00
			KIPV	0.009	0.09	0.00	0.00	0.00	0.05

Notes: The left column indicates the index of the cluster in $\mathcal{T}_R$, and in parentheses the most dominant protein class in it. Given this protein cluster, the second column indicates its most probable feature cluster, $t_F^* = \text{argmax}_{t_F} q(t_F \mid t_R)$. The next column indicates the probability of this feature cluster, given the protein cluster. Results are presented only when this value is greater than 0.8, indicating high coupling between both clusters. We further sort all features by $q(F \mid t_F^*)$ 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 proteins that contain this feature, normalized by the total number of proteins in the class). Clearly, the extracted features are correlated with the biological class associated with t_R.

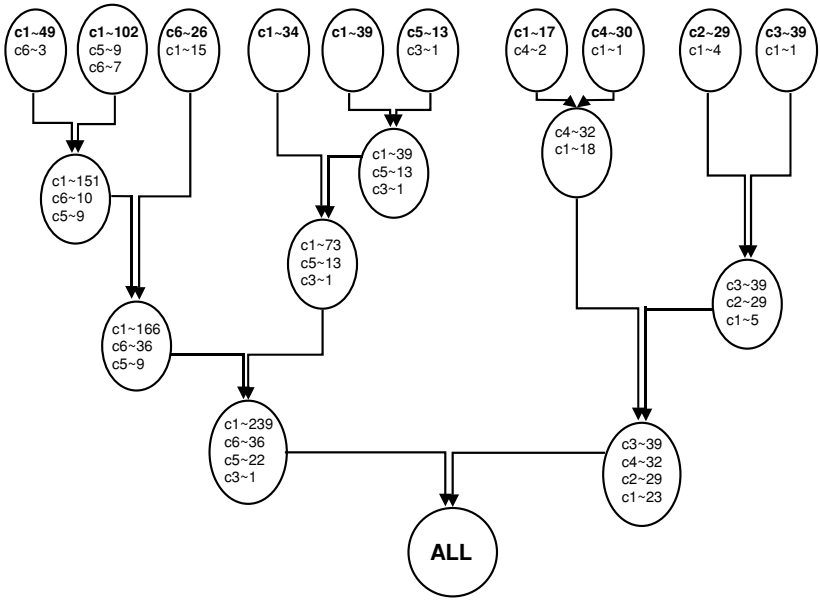

Figure 9: Application of the symmetric aIB to the GST protein data set. The learned protein cluster hierarchy, T_R, is presented from $|\mathcal{T}_R| = 10$ and below. In each cluster, the number of proteins from every class is indicated. For example, in the extreme upper right cluster, there are 39 proteins from the class c_3 and a single protein from the unlabeled class, c_1. After completing the experiments, we found that 36 out of the unlabeled (c_1) proteins were recently labeled as *Omega* class. This class is denoted by c_6 in the figure. Notice that all its members were clustered in the three left-most clusters.

Thus, our unsupervised analysis finds clusters that highly correlate with a manual partitioning of the proteins and simultaneously extracts features (subsequences of amino acids) that are good indicators for each such class.

Last, we apply the symmetric aIB to the same data. For comparison purposes, we consider the solution at $|\mathcal{T}_R| = |\mathcal{T}_F| = 10$. Here, $I(T_R; T_F) = 0.9$ *nats*, clearly inferior to the sIB result. However, the differences are mainly in the feature clusters, T_F, while the protein clusters obtained by aIB strongly correlate with the corresponding sIB solution, and thus also correlate with the protein class labels. In Figure 9 we present the T_R hierarchy.

Notice that many of our clusters correspond to the "unlabeled" c_1 class, and thus presumably correlate with additional—yet unknown—subclasses 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 (Apweiler et al., 2000). Thirty-six proteins out of this new *Omega* class were present in our data. In Figure 9 we see

that all these proteins were assigned automatically to a single branch in our hierarchy.

6.3 Triplet IB Application. We consider the specification of G_{in} and $G_{out}^{(a)}$ of Figure 2C, and use the triplet sIB algorithm over a simple text processing example.

6.3.1 Triplet sIB for Natural Language Processing. Our data consisted of seven Tarzan books by E. R. Burroughs, from which we got a sequence of about 600,000 words. We defined three random variables, W_p, W, and W_n, corresponding to the previous, current, and the next word in the sequence, respectively. For simplicity, we defined W as the set of 10 most frequent words in our data that are not neutral ("stop") words. Hence, we considered only word triplets in which the middle word was one of these 10 and defined $p(w_p, w, w_n)$ as the relative frequency of a triplet $\{w_p - w - w_n\}$ among all these triplets.

Given $p(w_p, w, w_n)$, 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, with $|\mathcal{T}_p| = 10$, $|\mathcal{T}_n| = 10$, and $\beta^{-1} = 0$, such that $I(T_p, T_n; W)$ is maximized.

In 50 different random initializations, the obtained solution always preserved more than 90% of the original information, $I(W_p, W_n; W) = 1.6\,nats$, although the dimensions of the compressed distribution $q(T_p, W, T_n)$ are more than 200 times smaller than those of the original matrix, $p(W_p, W, W_n)$. The best solution preserved about 94% of the original information, and we further concentrate on this solution. For every $w \in \mathcal{W}$, a manual examination of the two clusters that maximize $q(t_p, w, t_n)$, indicates that they consist of word pairs that are indicative of the word in between them, reflecting how T_p and T_n preserve the information about W (data not shown).

To validate the predictive power of T_p and T_n about W, we examined another book by E. R. Burroughs (*The Son of Tarzan*), which was not used while estimating $p(W_p, W, W_n)$ and constructing T_p and T_n. In this book, for every occurrence of one of the 10 words in $\mathcal{W}$, we try to predict it using its two neighbors, w_p and w_n. Specifically, w_p and w_n correspond to two clusters, $t_p \in \mathcal{T}_p$, $t_n \in \mathcal{T}_n$; thus, we predict the in-between word to be $\hat{w} = \text{argmax}_w q(w \mid t_p, t_n)$. For comparison, we also predict the in-between word from the complete joint statistics, $\hat{w} = \text{argmax}_w p(w \mid w_p, w_n)$, and while using a single neighbor, $\hat{w} = \text{argmax}_w p(w \mid w_p)$, and $\hat{w} = \text{argmax}_w p(w \mid w_n)$.

In Table 7 we present the precision and recall (Sebastiani, 2002) for all these prediction schemes. In spite of the significant compression implied by T_p and T_n, the averaged precision of their predictions is similar to those obtained using the original complete joint statistics. In terms of recall, predictions from the triplet IB clusters are even superior to those using the original W_p, W_n variables, since using $q(T_p, W, T_n)$ allows us to make predictions even for triplets that were not included in our training data.

Table 7: Precision and Recall Results for Triplet sIB.

	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

Notes: 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, that is, $q(W \mid T_p, T_n)$. The third column presents the precision while using the original joint statistics, that is, $p(W \mid W_p, W_n)$. The next two columns present the precision while using only one word neighbor for the prediction, that is, $p(W \mid W_p)$ 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 microaveraged precision and recall.

We note that this type of application might be useful in tasks like speech recognition, optical character recognition, and more, where it is not feasible to use the original joint distribution due to its high dimensionality. The triplet IB clusters provide a reasonable alternative that is dramatically less demanding. For biological sequence data, the analysis demonstrated here might be useful to gain further insights about the data properties.

7 Discussion and Future Research

7.1 An Information-Theoretic Perspective. The traditional approach to the analysis and modeling of empirical data is through generative models and maximum likelihood parameter estimation. However, complex data sets rarely come with their "correct" parametric model. Thus, the choice of the parametric class often involves nonobvious assumptions about the data. Shannon's (1948) information theory represents a radically different approach, which is concerned with two fundamental trade-offs. The first is between compression and distortion and is known as rate distortion theory or (lossy) source coding. The second is between reliable error correction and its cost and is known as the capacity-cost trade-off or channel coding (Cover & Thomas, 1991).

These two problems are dual components of one larger problem: the trade-off between distortion and cost, that is, the minimum average cost that is required for communication with at most a given average distortion. Shannon was able to break this problem, in the point-to-point communication setting, into a rate-distortion trade-off on one hand and a capacity-cost trade-off on the other. In the former, one minimizes mutual information subject to an average distortion constraint, while in the latter, one maximizes information subject to an average cost constraint. The optimal solution to both problems requires only the specification of the distortion or cost functions, without any assumptions about the nature of the underlying distributions, although they should be accessible. This is in sharp contrast to the statistical modeling approach. Nonetheless, in this work, we use a fundamental concept in the statistical modeling literature, known as Bayesian networks, in order to extend the information-theoretic trade-offs paradigm.

In the original IB, we balance between losing irrelevant distinctions made by X, while maintaining those that are relevant about Y. The first part—minimizing $I(T; X)$—is analogous to rate distortion, while the second part—maximizing $I(T; Y)$—is analogous to channel coding. Tishby et al. (1999) formulated both parts in a single principle and characterized its general solution. Here, we described a natural extension of these ideas, which allows the consideration of more complicated scenarios. This extension required an elevation of the point-to-point communication problem that lies behind the original IB to a network communication setting, a notoriously more difficult and largely unsolved problem. Nonetheless, as we demonstrated, such an extension is both viable and natural. In particular, the source-channel separation that exists in the original IB no longer holds nor is needed, as G_{in} and G_{out} replace the source and the channel terms of the original IB, respectively. It is thus possible that the current work will provide a way around the difficult multiterminal information theory problem, which remains unsolved to date (Cover & Thomas, 1991).

7.2 Finite Data and Sample Complexity Issues. An immediate obstacle in our information-theoretic approach is the assumption that the joint distribution, $p(\mathbf{X})$, is known, as typically we are given only a sample out of this distribution. Several ideas were suggested in this context for the original IB problem and are worth mentioning.

First, finite sample effects become more severe as the complexity of the new representation, $\mathcal{T}$, increases. That is, overfitting is more evident as the number of clusters increases (Pereira et al., 1993; Still & Bialek, 2003). Thus, in particular, one should apply with great care agglomerative algorithms that begin with substantial overfitting for small samples. Having said that, it is worth noting that the agglomerative IB algorithm is supported through the work of Gutman (1989). Specifically, theorem 1 in that work implies

that the *JS* divergence, used here to determine which clusters to merge, is the optimal criterion to decide whether two empirical samples came from a single source.

Next, given the empirical evidence, it might be useful to use the least informative distribution as the input to the IB analysis. The notion of least informative distributions, under expectation constraints, is related to maximum entropy methods and has been used recently for dimension reduction (Globerson & Tishby, 2004). One can prove generalization sample complexity bounds for the IB problem within this framework.

Finally, it was shown that one can introduce a corrected IB functional in which finite sample effects are taken directly into account for the relevant information term (Still & Bialek, 2003; Atwal & Slonim, 2005).

Extending these works in our context is clearly important yet out of the scope of this work.

7.3 Future Research

7.3.1 Choosing the Number of Clusters. A natural question in cluster analysis is the estimation of the "correct" number of clusters. It is important to bear in mind, though, that this question might have more than one proper answer, for example, if there is a natural hierarchical structure in $p(\mathbf{X})$ such that different resolutions convey multiple important insights.

The number of clusters we extract is related to the trade-off parameter β, where low β values imply a relatively low resolution, while high β values suggest that a large number of clusters should be employed. The deterministic annealing multivariate IB algorithm seems to be most relevant here since it automatically adjusts the resolutions of the different clustering systems as β is increased. For the original IB problem, a recent rigorous treatment characterizes the maximal β value that can be employed for given data before overfit effects take place (Still & Bialek, 2003). Extending this work in our context is a natural direction for future research.

7.3.2 Relation to Other Methods and Parametric Multivariate IB. The possible connections with other data analysis methods merit further investigation. The general structure of the multivariate iIB algorithm is reminiscent of EM (Dempster, Laird, & Rubin, 1977). Moreover, there are strong relations between the original IB problem and maximum likelihood estimation for mixture models (Slonim & Weiss, 2002). Hence, it is natural to look for further relationships between generative models and different multivariate IB problems. In particular, formulating new problems with the multivariate IB framework 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) (Bell & Sejnowski, 1995). The parallel IB provides an ICA-like decomposition with

an important distinction. In contrast to ICA, it is aimed at preserving information about predefined aspects of the data, specified through the choice of the relevant variables in G_{out}.

A parametric variant of our framework might be useful in different situations (Elidan & Friedman, 2003). This issue seem to be better addressed by the structural multivariate IB principle, $\mathcal{L}^{(2)}$. In this formulation, we aim to minimize the *KL* divergence between $q(\mathbf{X}, \mathbf{T})$ and the target class defined by G_{out}. If we further require a particular parametric form over this class, minimizing this *KL* corresponds to finding $q(\mathbf{X}, \mathbf{T})$ with minimum violation of the conditional independencies implied by G_{out} and with the appropriate parametric form. In particular, this means that the number of free parameters can be drastically reduced, avoiding possible redundant solutions.

7.3.3 Multivariate Relevance Compression Function. In the original IB problem, the trade-off in the IB functional is quantified by the relevance-compression function (also known as the information curve). Given $p(X, Y)$, this concave function bounds the maximal achievable $I(T; Y)$, for any level of compression, $I(T; X)$ (Gilad-Bachrah, Navot, & Tishby, 2003). It is intimately related to the rate distortion function and the capacity cost function, and in a sense unifies them, as discussed earlier. It depends solely on $p(X, Y)$ and characterizes the structure in this joint distribution: the clearer this structure is, the steeper this curve becomes.

Analogously, given $p(\mathbf{X})$, we may consider the multivariate relevance compression function as the two-dimensional optimal curve of the maximally attainable $\mathcal{I}^{G_{out}}$ for any level of $\mathcal{I}^{G_{in}}$. From the variational principle, $\mathcal{L}^{(1)}$, it follows that the slope of this curve is β^{-1}. Thus, assuming that this curve is differentiable, it must be downward concave, as in the original IB case. Importantly, though, this function depends on the specification of G_{in} and G_{out}. That is, given $p(\mathbf{X})$, there are many different such functions that characterize the structure in this joint distribution in multiple ways.

7.3.4 How to specify G_{in} and G_{out}. The underlying assumption in our formulation is that G_{in} and G_{out} are provided as part of the problem setup. However, 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 k can be seen as a model selection task, and certainly not an easy one.

Thus, an important issue is to develop automatic methods for specifying both networks. Possible guidance can come from the multivariate relevance compression function. Specifically, it seems plausible to prefer specifications that yield steeper relevance compression curves. This issue clearly deserves further investigation.

7.4 Conclusion. Our formulation corresponds to a rich family of optimization problems that are all unified under the same information-theoretic

framework. In particular, it allows one to extract structure from data in many different ways. In this work, we focused on three examples: parallel IB, symmetric IB, and triplet IB. However, we believe that this is only the tip of the iceberg.

An immediate corollary of our analysis is that the general term of *clustering* conceals a broad family of many distinct problems that deserve special consideration. To the best of our knowledge, the multivariate IB framework described in this work is the first successful attempt to define these subproblems, solve them, and demonstrate their importance.

Appendix A: Proofs

Proof of Proposition 1. Using the multi-information definition in equation 3.1 and the fact that $p(\mathbf{X}) \models G$, we get

$$\begin{aligned}
\mathcal{I}(\mathbf{X}) &= E_{p(\mathbf{x})}\left[\log \frac{p(\mathbf{x})}{p(x_1)\dots p(x_n)}\right] \\
&= E_{p(\mathbf{x})}\left[\log \Pi_{i=1}^{n} \frac{p\left(x_i \mid \mathbf{pa}_{X_i}^{G}\right)}{p(x_i)}\right] \\
&= \sum_{i=1}^{n} E_{p(\mathbf{x})}\left[\log \frac{p\left(x_i \mid \mathbf{pa}_{X_i}^{G}\right)}{p(x_i)}\right] \\
&= \sum_{i=1}^{n} I\left(X_i; \mathrm{Pa}_{X_i}^{G}\right).
\end{aligned}$$

Proof of Proposition 2.

$$\begin{aligned}
D_{KL}[p\|G] &= \min_{q\models G} E_{p(\mathbf{x})}\left[\log \frac{p(x_1,\dots,x_n)}{q(x_1,\dots,x_n)}\right] \\
&= \min_{q\models G} E_{p(\mathbf{x})}\left[\log \frac{p(x_1,\dots,x_n)}{\Pi_{i=1}^{n} p\left(x_i \mid \mathbf{pa}_{X_i}^{G}\right)}\right] \\
&\quad + E_{p(\mathbf{x})}\left[\log \frac{\Pi_{i=1}^{n} p\left(x_i \mid \mathbf{pa}_{X_i}^{G}\right)}{\Pi_{i=1}^{n} q\left(x_i \mid \mathbf{pa}_{X_i}^{G}\right)}\right] \\
&= E_{p(\mathbf{x})}\left[\log \frac{p(x_1,\dots,x_n)}{\Pi_{i=1}^{n} p\left(x_i \mid \mathbf{pa}_{X_i}^{G}\right)}\right] \\
&\quad + \min_{q\models G}\left[\sum_{i=1}^{n} \sum_{x_i,\mathbf{pa}_{X_i}^{G}} p\left(\mathbf{pa}_{X_i}^{G}\right) p\left(x_i \mid \mathbf{pa}_{X_i}^{G}\right) \log \frac{p\left(x_i \mid \mathbf{pa}_{X_i}^{G}\right)}{q\left(x_i \mid \mathbf{pa}_{X_i}^{G}\right)}\right]
\end{aligned}$$

$$= E_{p(\mathbf{x})}\left[\log \frac{p(x_1,\ldots,x_n)}{\Pi_{i=1}^n p\left(x_i \mid \mathbf{pa}_{X_i}^G\right)}\right]$$

$$+ \min_{q \models G}\left[\sum_{i=1}^n \sum_{\mathbf{pa}_{X_i}^G} p\left(\mathbf{pa}_{X_i}^G\right) D_{KL}\left[p\left(x_i \mid \mathbf{pa}_{X_i}^G\right) \| q\left(x_i \mid \mathbf{pa}_{X_i}^G\right)\right]\right],$$

and since the right term is nonnegative 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 3. We use proposition 2:

$$\begin{aligned}
D_{KL}[p\|G] &= \min_{q \models G} E_{p(\mathbf{x})}\left[\log \frac{p(x_1,\ldots,x_n)}{q(x_1,\ldots,x_n)}\right] \\
&= E_{p(\mathbf{x})}\left[\log \frac{p(x_1,\ldots,x_n)}{\Pi_{i=1}^n p\left(x_i \mid \mathbf{pa}_{X_i}^G\right)}\right] \\
&= E_{p(\mathbf{x})}\left[\log \frac{\Pi_{i=1}^n p(x_i \mid x_1,\ldots,x_{i-1})}{\Pi_{i=1}^n p\left(x_i \mid \mathbf{pa}_{X_i}^G\right)}\right] \\
&= \sum_{i=1}^n E_{p(\mathbf{x})}\left[\log \frac{p(x_i \mid x_1,\ldots,x_{i-1})}{p\left(x_i \mid \mathbf{pa}_{X_i}^G\right)}\right] \\
&= \sum_{i=1}^n I\left(X_i;\ \{X_1,\ldots,X_{i-1}\} \setminus \mathbf{Pa}_{X_i}^G \mid \mathbf{Pa}_{X_i}^G\right),
\end{aligned}$$

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

$$\begin{aligned}
D_{KL}[p\|G] &= E_{p(\mathbf{x})}\left[\log \frac{p(x_1,\ldots,x_n)}{\Pi_{i=1}^n p\left(x_i \mid \mathbf{pa}_{X_i}^G\right)}\right] \\
&= E_{p(\mathbf{x})}\left[\log \frac{p(x_1,\ldots,x_n)}{\Pi_{i=1}^n p(x_i)}\right] - E_{p(\mathbf{x})}\left[\log \frac{\Pi_{i=1}^n p\left(x_i \mid \mathbf{pa}_{X_i}^G\right)}{\Pi_{i=1}^n p(x_i)}\right] \\
&= \mathcal{I}(\mathbf{X}) - \sum_{i=1}^n E_{p(\mathbf{x})}\left[\log \frac{p\left(x_i \mid \mathbf{pa}_{X_i}^G\right)}{p(x_i)}\right] \\
&= \mathcal{I}(\mathbf{X}) - \sum_{i=1}^n I\left(X_i;\ \mathbf{Pa}_{X_i}^G\right).
\end{aligned}$$

Proof of Theorem 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 to get the Lagrangian,

$$\tilde{\mathcal{L}}[q(\mathbf{X},\mathbf{T})] = \sum_{\ell=1}^{k} I(T_\ell;\mathbf{U}_\ell) - \beta\left[\sum_{i=1}^{n} I(X_i;\mathbf{V}_{X_i}) + \sum_{\ell=1}^{k} I(T_\ell;\mathbf{V}_{T_\ell})\right] + \sum_{\mathbf{u}_\ell}\lambda(\mathbf{u}_\ell)\sum_{t_\ell} q(t_\ell \mid \mathbf{u}_\ell), \tag{A.1}$$

where we drop terms that depend only on the observed variables $\mathbf{X}$. To differentiate $\tilde{\mathcal{L}}$ with respect to a specific parameter $q(t_j \mid \mathbf{u}_j)$ we use the following two lemmas. In the proofs of these two lemmas, we assume that $q(\mathbf{X},\mathbf{T}) \models G_{in}$ and that the $\mathbf{T}$ variables are all leafs in G_{in}.

Lemma 1. *Under the above normalization constraints, for every event $\boldsymbol{a}$ over $\boldsymbol{X} \cup \boldsymbol{T}$ (that is, $\boldsymbol{a}$ is some assignment to some subset of $\boldsymbol{X} \cup \boldsymbol{T}$), we have*

$$\frac{\partial q(\boldsymbol{a})}{\partial q(t_j \mid \boldsymbol{u}_j)} = q(\boldsymbol{u}_j)q(\boldsymbol{a} \mid t_j, \boldsymbol{u}_j). \tag{A.2}$$

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 $\sum_{\mathbf{z},\mathbf{a}} q(\mathbf{x},\mathbf{t})$ means that the sum is only over the variables in $\mathbf{Z}$ (where the others are set through $\mathbf{a}$):

$$\begin{aligned}\frac{\partial q(\mathbf{a})}{\partial q(t_j \mid \mathbf{u}_j)} &= \frac{\partial}{\partial q(t_j \mid \mathbf{u}_j)}\sum_{\mathbf{z},\mathbf{a}} q(\mathbf{x},\mathbf{t}) \\ &= \frac{\partial}{\partial q(t_j \mid \mathbf{u}_j)}\sum_{\mathbf{z},\mathbf{a}} \Pi_{\ell=1}^{k} q(t_\ell \mid \mathbf{u}_\ell) \\ &= \sum_{\mathbf{z},\mathbf{a}} \frac{\partial}{\partial q(t_j \mid \mathbf{u}_j)}\Pi_{\ell=1}^{k} q(t_\ell \mid \mathbf{u}_\ell).\end{aligned}$$

Clearly the derivatives are nonzero only for terms in which $t_l = t_j$ and $\mathbf{u}_l = \mathbf{u}_j$. For each such term, the derivative is simply $\Pi_{\ell=1,\ell\neq j}^{k} q(t_\ell \mid \mathbf{u}_\ell)$. Dividing and multiplying every such term by $q(t_j \mid \mathbf{u}_j)$, we obtain

$$\begin{aligned}\frac{\partial q(a)}{\partial q(t_j \mid \mathbf{u}_j)} &= \frac{1}{q(t_j \mid \mathbf{u}_j)}\sum_{\mathbf{z}\setminus\{t_j,\mathbf{u}_j\},a,t_j,\mathbf{u}_j} \Pi_{\ell=1}^{k} q(t_\ell \mid \mathbf{u}_\ell) \\ &= \frac{q(a,t_j,\mathbf{u}_j)}{q(t_j \mid \mathbf{u}_j)}\end{aligned}$$

$$= q(\mathbf{u}_j) q(a \mid t_j, \mathbf{u}_j).$$

Using this lemma, we get:

Lemma 2. *For every Y, $\boldsymbol{Z} \subseteq \boldsymbol{X} \cup \boldsymbol{T}$*

$$\frac{\partial I(Y; \boldsymbol{Z})}{\partial q(t_j \mid \boldsymbol{u}_j)} = q(\boldsymbol{u}_j) \sum_{y,z} \left[q(y, z \mid t_j, \boldsymbol{u}_j) log \frac{q(y \mid z)}{q(y)} - 1 \right]. \tag{A.3}$$

Proof.

$$\begin{aligned}
\frac{\partial I(Y; \mathbf{Z})}{\partial q(t_j \mid \mathbf{u}_j)} = & \sum_{\mathbf{y},\mathbf{z}} \log \frac{q(\mathbf{y} \mid \mathbf{z})}{q(\mathbf{y})} \frac{\partial}{\partial q(t_j \mid \mathbf{u}_j)} q(\mathbf{y}, \mathbf{z}) \\
& + \sum_{\mathbf{y},\mathbf{z}} \frac{\partial}{\partial q(t_j \mid \mathbf{u}_j)} q(\mathbf{y}, \mathbf{z}) \\
& - \sum_{\mathbf{y},\mathbf{z}} q(\mathbf{z} \mid \mathbf{y}) \frac{\partial}{\partial q(t_j \mid \mathbf{u}_j)} q(\mathbf{y}) \\
& - \sum_{\mathbf{y},\mathbf{z}} q(\mathbf{y} \mid \mathbf{z}) \frac{\partial}{\partial q(t_j \mid \mathbf{u}_j)} q(\mathbf{z}).
\end{aligned}$$

Applying lemma 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 equation A.1. Note that we can ignore terms that do not depend on the value of T_j since these are constants with respect to $q(t_j \mid \mathbf{u}_j)$. Therefore, by taking the derivative and equating to zero, we get:

$$\begin{aligned}
\log q(t_j \mid \mathbf{u}_j) = & \log q(t_j) \\
& - \beta \Bigg[\sum_{i: T_j \in \mathbf{V}_{X_i}} \sum_{\mathbf{v}_{X_i}^{-j}, x_i} q\left(\mathbf{v}_{X_i}^{-j} \mid \mathbf{u}_j\right) q\left(x_i \mid \mathbf{v}_{X_i}^{-j}, \mathbf{u}_j\right) \log \frac{p(x_i)}{q\left(x_i \mid \mathbf{v}_{X_i}^{-j}, t_j\right)} \\
& - \sum_{\ell: T_j \in \mathbf{V}_{T_\ell}} \sum_{\mathbf{v}_{T_\ell}^{-j}, t_\ell} q\left(\mathbf{v}_{T_\ell}^{-j} \mid \mathbf{u}_j\right) q\left(t_\ell \mid \mathbf{v}_{T_\ell}^{-j}, \mathbf{u}_j\right) \log \frac{q(t_\ell)}{q\left(t_\ell \mid \mathbf{v}_{T_\ell}^{-j}, t_j\right)} \\
& - \sum_{\mathbf{v}_{T_j}} q(\mathbf{v}_{T_j} \mid \mathbf{u}_j) \log \frac{q(\mathbf{v}_{T_j})}{q(\mathbf{v}_{T_j} \mid t_j)} \Bigg] + c(\mathbf{u}_j),
\end{aligned} \tag{A.4}$$

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

$$\sum_{\mathbf{v}_{X_i}^{-j}, X_i} q\left(\mathbf{v}_{X_i}^{-j} \mid \mathbf{u}_j\right) q\left(x_i \mid \mathbf{v}_{X_i}^{-j}, \mathbf{u}_j\right) \log \frac{q\left(x_i \mid \mathbf{v}_{X_i}^{-j}, \mathbf{u}_j\right)}{p(x_i)}, \tag{A.5}$$

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 on the right-hand side of equation A.4. Hence, we end up with

$$\begin{aligned}
\log q(t_j \mid \mathbf{u}_j) = \log q(t_j) \\
&-\beta \cdot \Bigg[\sum_{i: T_j \in \mathbf{V}_{X_i}} E_{q(\cdot \mid \mathbf{u}_j)} \Big[D_{KL}\big[q\left(X_i \mid \mathbf{v}_{X_i}^{-j}, \mathbf{u}_j\right) \| q\big(X_i \mid \mathbf{v}_{X_i}^{-j}, t_j\big)\big] \Big] \\
&- \sum_{\ell: T_j \in \mathbf{V}_{T_\ell}} E_{q(\cdot \mid \mathbf{u}_j)} \Big[D_{KL}\big[q\left(T_\ell \mid \mathbf{v}_{T_\ell}^{-j}, \mathbf{u}_j\right) \| q\left(T_\ell \mid \mathbf{v}_{T_\ell}^{-j}, t_j\right)\big] \Big] \\
&- D_{KL}\left[q(\mathbf{V}_{T_j} \mid \mathbf{u}_j) \| q(\mathbf{V}_{T_j} \mid t_j)\right] \Bigg] + c(\mathbf{u}_j).
\end{aligned} \tag{A.6}$$

Finally, taking the exponent and applying the normalization constraints for each distribution $q(t_j \mid \mathbf{u}_j)$ completes the proof.

Proof of Theorem 2. The following lemma allows drawing the connection between T_j and the other variables after every merger.

Lemma 3. *Let* $\boldsymbol{Y}, \boldsymbol{Z} \subset \boldsymbol{X} \cup \boldsymbol{T}$. *Then*

$$q\left(\boldsymbol{z}, \bar{t}_j\right) = q\left(\boldsymbol{z}, t_j^{\ell}\right) + q\left(\boldsymbol{z}, t_j^{r}\right), \tag{A.7}$$

and

$$q(\boldsymbol{y} \mid \boldsymbol{z}, \bar{t}_j) = \pi_{\ell,\boldsymbol{z}} \cdot q\left(\boldsymbol{y} \mid \boldsymbol{y}, t_j^{\ell}\right) + \pi_{r,\boldsymbol{z}} \cdot q\left(\boldsymbol{y} \mid \boldsymbol{z}, t_j^{r}\right). \tag{A.8}$$

Proof. 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 $\mathbf{W} = \emptyset$. For every $\mathbf{z}$, $\bar{t}_j$ we have

$$\begin{aligned}
q(\mathbf{z}, \bar{t}_j) &= q(\mathbf{z}) p(\bar{t}_j \mid \mathbf{z}) \\
&= q(\mathbf{z}) \sum_{\mathbf{u}_j{}^{-\mathbf{w}}} q\left(\mathbf{u}_j{}^{-\mathbf{w}} \mid \mathbf{z}\right) q\left(\bar{t}_j \mid \mathbf{z}^{-\mathbf{w}}, \mathbf{w}, \mathbf{u}_j{}^{-\mathbf{w}}\right) \\
&= q(\mathbf{z}) \sum_{\mathbf{u}_j{}^{-\mathbf{w}}} q\left(\mathbf{u}_j{}^{-\mathbf{w}} \mid \mathbf{z}\right) q\left(\bar{t}_j \mid \mathbf{u}_j\right),
\end{aligned}$$

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 equation 5.5, we find that

$$\begin{aligned} q(\mathbf{z}, \bar{t}_j) &= q(\mathbf{z}) \sum_{\mathbf{u}_j{}^{-\mathbf{w}}} q\left(\mathbf{u}_j{}^{-\mathbf{w}} \mid \mathbf{z}\right) \left(q\left(t_j^\ell \mid \mathbf{u}_j\right) + q\left(t_j^r \mid \mathbf{u}_j\right)\right) \\ &= q(\mathbf{z}) \sum_{\mathbf{u}_j{}^{-\mathbf{w}}} q\left(\mathbf{u}_j{}^{-\mathbf{w}} \mid \mathbf{z}\right) \left(q\left(t_j^\ell \mid \mathbf{z}^{-\mathbf{w}}, \mathbf{w}, \mathbf{u}_j{}^{-\mathbf{w}}\right) + q\left(t_j^r \mid \mathbf{z}^{-\mathbf{w}}, \mathbf{w}, \mathbf{u}_j{}^{-\mathbf{w}}\right)\right), \end{aligned}$$

where again we used the structure of G_{in}. Since $\mathbf{Z} = \mathbf{Z}^{-\mathbf{W}} \cup \{\mathbf{W}\}$, we get

$$\begin{aligned} q(\mathbf{z}, \bar{t}_j) &= q(\mathbf{z}) \sum_{\mathbf{u}_j{}^{-\mathbf{w}}} \left(q\left(\mathbf{u}_j{}^{-\mathbf{w}}, t_j^\ell \mid \mathbf{z}\right) + q\left(\mathbf{u}_j{}^{-\mathbf{w}}, t_j^r \mid \mathbf{z}\right)\right) \\ &= q\left(\mathbf{z}, t_j^\ell\right) + q\left(\mathbf{z}, t_j^r\right), \end{aligned}$$

as required. To prove the second part we first note that if $q(\mathbf{z}, \bar{t}_j) = 0$, then both sides of equation A.8 are trivially equal; thus, we assume that this is not the case. Then, for every $\mathbf{y}$, $\mathbf{z}$, $\bar{t}_j$, we have

$$\begin{aligned} q(\mathbf{y} \mid \mathbf{z}, \bar{t}_j) &= \frac{q(\mathbf{y}, \mathbf{z}, \bar{t}_j)}{q(\mathbf{z}, \bar{t}_j)} \\ &= \frac{q\left(\mathbf{y}, \mathbf{z}, t_j^\ell\right) + q\left(\mathbf{y}, \mathbf{z}, t_j^r\right)}{q\left(\mathbf{z}, \bar{t}_j\right)} \\ &= \frac{q\left(t_j^\ell \mid \mathbf{z}\right)}{q\left(\bar{t}_j \mid \mathbf{z}\right)} q\left(\mathbf{y} \mid \mathbf{z}, t_j^\ell\right) + \frac{q\left(t_j^r \mid \mathbf{z}\right)}{q\left(\bar{t}_j \mid \mathbf{z}\right)} q\left(\mathbf{y} \mid \mathbf{z}, t_j^r\right). \end{aligned}$$

Hence from equation 5.6, we get the desired form.

Next, we need the following 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 4. *The reduction of the mutual information $I(\boldsymbol{Y}; \boldsymbol{V})$ due to the merger $\{t_j^\ell, t_j^r\} \Rightarrow \bar{t}_j$ is given by*

$$\begin{aligned} \Delta I(\boldsymbol{Y}; \boldsymbol{V}) &\equiv I(\boldsymbol{Y}; \boldsymbol{V}^{bef}) - I(\boldsymbol{Y}; \boldsymbol{V}^{afT}) \\ &= q(\bar{t}_j) \cdot E_{q(\cdot \mid \bar{t}_j)}[JS_{\Pi_{v^{-j}}}[q(\boldsymbol{Y} \mid t_j^\ell, \boldsymbol{v}^{-j}), q(\boldsymbol{Y} \mid t_j^r, \boldsymbol{v}^{-j})]]. \end{aligned}$$

Proof. Using the chain rule for mutual information (Cover & Thomas, 1991, p. 22), we get

$$\begin{aligned}\Delta I(\mathbf{Y};\mathbf{V}) &= I(\mathbf{V}^{-j};\mathbf{Y}) + I\left(T_j^{bef};\mathbf{Y} \mid \mathbf{V}^{-j}\right) - I(\mathbf{V}^{-j};\mathbf{Y}) - I\left(T_j^{aft};\mathbf{Y} \mid \mathbf{V}^{-j}\right) \\ &= I\left(T_j^{bef};\mathbf{Y} \mid \mathbf{V}^{-j}\right) - I\left(T_j^{aft};\mathbf{Y} \mid \mathbf{V}^{-j}\right).\end{aligned}$$

From equation 5.5, we find that

$$\Delta I(\mathbf{Y};\mathbf{V}) = \sum_{\mathbf{v}^{-j}} q(\mathbf{v}^{-j}) \Delta I(\mathbf{v}^{-j}),$$

where we used the notation

$$\begin{aligned}\Delta I(\mathbf{v}^{-j}) = &\sum_{\mathbf{y}} q\left(t_j^{\ell},\mathbf{y} \mid \mathbf{v}^{-j}\right) \log \frac{q\left(\mathbf{y} \mid t_j^{\ell},\mathbf{v}^{-j}\right)}{q\left(\mathbf{y} \mid \mathbf{v}^{-j}\right)} \\ &+ \sum_{\mathbf{y}} q\left(t_j^{r},\mathbf{y} \mid \mathbf{v}^{-j}\right) \log \frac{q\left(\mathbf{y} \mid t_j^{r},\mathbf{v}^{-i}\right)}{q\left(\mathbf{y} \mid \mathbf{v}^{-j}\right)} \\ &- \sum_{\mathbf{y}} q\left(\bar{t}_j,\mathbf{y} \mid \mathbf{v}^{-j}\right) \log \frac{q\left(\mathbf{y} \mid \bar{t}_j,\mathbf{v}^{-i}\right)}{q\left(\mathbf{y} \mid \mathbf{v}^{-j}\right)}.\end{aligned}$$

Using lemma 3 (with $\mathbf{Z} = \mathbf{Y} \cup \mathbf{V}^{-j}$), we obtain

$$q\left(\bar{t}_j,\mathbf{y} \mid \mathbf{v}^{-j}\right) = q\left(t_j^{\ell},\mathbf{y} \mid \mathbf{v}^{-j}\right) + q\left(t_j^{r},\mathbf{y} \mid \mathbf{v}^{-j}\right).$$

Setting this in the previous equation, we get,

$$\begin{aligned}\Delta I(\mathbf{v}^{-j}) = &\sum_{\mathbf{y}} q\left(t_j^{\ell},\mathbf{y} \mid \mathbf{v}^{-j}\right) \log \frac{q\left(\mathbf{y} \mid t_j^{\ell},\mathbf{v}^{-j}\right)}{q\left(\mathbf{y} \mid \bar{t}_j,\mathbf{v}^{-j}\right)} \\ &+ \sum_{\mathbf{y}} q\left(t_j^{r},\mathbf{y} \mid \mathbf{v}^{-j}\right) \log \frac{q\left(\mathbf{y} \mid t_j^{r},\mathbf{v}^{-j}\right)}{q\left(\mathbf{y} \mid \bar{t}_j,\mathbf{v}^{-j}\right)} \\ = &\, q\left(\bar{t}_j \mid \mathbf{v}^{-j}\right) \cdot \pi_{\ell,\mathbf{v}^{-j}} \sum_{\mathbf{y}} q\left(\mathbf{y} \mid t_j^{\ell},\mathbf{v}^{-j}\right) \log \frac{q\left(\mathbf{y} \mid t_j^{\ell},\mathbf{v}^{-j}\right)}{q\left(\mathbf{y} \mid \bar{t}_j,\mathbf{v}^{-j}\right)} \\ &+ q\left(\bar{t}_j \mid \mathbf{v}^{-j}\right) \cdot \pi_{r,\mathbf{v}^{-j}} \sum_{\mathbf{y}} q\left(\mathbf{y} \mid t_j^{r},\mathbf{v}^{-j}\right) \log \frac{q\left(\mathbf{y} \mid t_j^{r},\mathbf{v}^{-j}\right)}{q\left(\mathbf{y} \mid \bar{t}_j,\mathbf{v}^{-j}\right)}.\end{aligned}$$

However, using again lemma 3, we see that

$$q\left(\mathbf{y} \mid \bar{t}_j, \mathbf{v}^{-j}\right) = \pi_{\ell,\mathbf{v}^{-j}} \cdot q\left(\mathbf{y} \mid t_j^{\ell}, \mathbf{v}^{-j}\right) + \pi_{r,\mathbf{v}^{-j}} \cdot q\left(\mathbf{y} \mid t_j^{r}, \mathbf{v}^{-j}\right).$$

Therefore, using the *JS* definition in equation 5.8, we get

$$\Delta I(\mathbf{v}^{-j}) = q\left(\bar{t}_j \mid \mathbf{v}^{-j}\right) \cdot JS_{\Pi_{\mathbf{v}^{-j}}}\left[q\left(\mathbf{Y} \mid t_j^{\ell}, \mathbf{v}^{-j}\right), q\left(\mathbf{Y} \mid t_j^{r}, \mathbf{v}^{-j}\right)\right].$$

Setting this back in the expression for $\Delta I(\mathbf{Y}; \mathbf{V})$, we get

$$\begin{aligned}\Delta I(\mathbf{Y}; \mathbf{V}) &= \sum_{\mathbf{v}^{-j}} q\left(\mathbf{v}^{-j}\right) q\left(\bar{t}_j \mid \mathbf{v}^{-j}\right) \cdot JS_{\Pi_{\mathbf{v}^{-j}}}\left[q\left(\mathbf{Y} \mid t_j^{\ell}, \mathbf{v}^{-j}\right), q\left(\mathbf{Y} \mid t_j^{r}, \mathbf{v}^{-j}\right)\right] \\ &= q\left(\bar{t}_j\right) \cdot E_{q(\cdot \mid \bar{t}_j)}\left[JS_{\Pi_{\mathbf{v}^{-j}}}\left[q\left(\mathbf{Y} \mid t_j^{\ell}, \mathbf{v}^{-j}\right), q\left(\mathbf{Y} \mid t_j^{r}, \mathbf{v}^{-j}\right)\right] \right].\end{aligned}$$

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,

$$\begin{aligned}\Delta\mathcal{L}\left(t_j^{\ell}, t_j^{r}\right) = &\sum_{i:T_j \in \mathbf{V}_{X_i}} \Delta I(X_i; \mathbf{V}_{X_i}) + \sum_{\ell:T_j \in \mathbf{V}_{T_\ell}} \Delta I(T_\ell; \mathbf{V}_{T_\ell}) + \Delta I(T_j; \mathbf{V}_{T_j}) \\ &-\beta^{-1}\Delta I(T_j; \mathbf{U}_j). \qquad \text{(A.9)}\end{aligned}$$

Applying lemma 4 for each of these information terms, we get the desired form.

Appendix B: Implementation and Preprocess Details

In this appendix we describe the details of the implementation and the preprocess applied in our examples. In several cases, in order to avoid too high dimensionality, we apply feature selection by information gain before the clustering is applied. Specifically, given a joint distribution, $p(X, Y)$, we sort all X values by their contribution to $I(X; Y)$: $p(x)\sum_y p(y \mid x) \log \frac{p(y|x)}{p(y)}$, and select only the top sorted values for further analysis.

Parallel sIB for Style Versus Topic Text Clustering

- All books were downloaded from the Gutenberg Project, http://promo.net/pg/.
- Uppercase characters were lowered, digits were united to one symbol, and nonalphanumeric characters were ignored.

- Each book was split into "documents" consisting of 200 successive words each, ending up with 1346 documents and 15,528 distinct words.
- After normalization, we got an estimated joint distribution, $p(D, W)$, where $p(D)$ is uniform and each entry indicates the probability that a random word position is equal to $w \in \mathcal{W}$ while the document is $d \in \mathcal{D}$.
- We applied the parallel sIB algorithm to this $p(D, W)$ with $\mathbf{T} = \{T_1, T_2\}$, $|\mathcal{T}_j| = 2$ and 5 different initializations per T_j.

Parallel sIB for Gene Expression Data Analysis

- We used the gene expression measurements of approximately 6800 human genes in 72 samples of leukemia (Golub et al., 1999).
- We removed about 1500 genes that were not expressed in the data and normalized the measurements of the remaining 5288 genes in each sample independently, to get an estimated joint distribution $p(S, G)$ over samples and genes, with uniform $p(S)$.
- We sorted all genes by their contribution to $I(S; G)$, and selected the 500 most informative ones.
- After renormalization of the measurements in each sample, we ended up with an estimated joint distribution, $p(S, G)$, with $|\mathcal{S}| = 72$, $|\mathcal{G}| = 500$, and $p(S) = \frac{1}{|\mathcal{S}|}$.
- We applied the parallel sIB algorithm to this $p(S, G)$ with $\mathbf{T} = \{T_1, \ldots, T_4\}$, $|\mathcal{T}_j| = 2$, and 5 different initializations per T_j.

Symmetric dIB and iIB for Word-Topic Clustering

- We used the 20-news-group corpus (Lang, 1995), which contains about 20,000 documents and messages, distributed among 20 discussion groups, or topics.
- We removed all file headers, lowered uppercase characters, united digits into one symbol, and ignored nonalphanumeric characters.
- We further removed stop words and words with only one occurrence, ending up with a counts matrix of $|\mathcal{D}| = 19,997$ documents versus $|\mathcal{W}| = 74,000$ unique words.
- By summing the word counts of all the documents in each topic and applying simple normalization, we extracted an estimated joint distribution, $p(W, C)$, of words versus topics with $|\mathcal{C}| = 20$.
- We sorted all words by their contribution to $I(W; C)$ and selected the 200 most informative ones. After renormalization, we ended up with a joint distribution with $|\mathcal{W}| = 200$, $|\mathcal{C}| = 20$.

- We applied the symmetric dIB algorithm to this joint distribution. Increasing β was done through $\beta^{(r+1)} = (1 + \varepsilon_\beta)\beta^{(r)}$, $\varepsilon_\beta = 0.001$; $\beta^{(0)} = \varepsilon_\beta$. The split detection parameters were $b = \frac{1}{\beta}$, that is, as β increases, the algorithm becomes more liberal for declaring cluster splits. The scaling factor for the stochastic duplication was $\alpha = 0.005$. However, before the first split, we used $\alpha_1 = 0.95$, so as to avoid the attractor of the trivial fixed point, $q(T_j \mid \mathbf{U}_j) = q(T_j)$.

Symmetric sIB and aIB for Protein Sequence Analysis

- Each protein was represented as a count vector over all the 38,421 different 4-mers of amino acids present in the data.
- After normalizing the counts for each protein independently, we got a joint distribution $p(R, F)$ with $p(R) = \frac{1}{|\mathcal{R}|}$. We sorted all features by their contribution to $I(R; F)$ and selected the top 2,000.
- After renormalization, we ended up with a joint distribution $p(R, F)$ with $|\mathcal{R}| = 421$, $|\mathcal{F}| = 2,000$ and $p(R) = \frac{1}{|\mathcal{R}|}$.
- In the sIB algorithm, we used for the initialization the strategy described in Slonim & Tishby (2000). We randomly initialize only T_F and optimize it using the original sIB algorithm (Slonim et al., 2002), such that $I(T_F; R)$ is maximized. Given this T_F, we randomly initialize T_R and use again the original 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 6 until convergence is attained.
- We repeat this procedure 100 times and select the solution that maximizes $I(T_R; T_F)$.

Triplet sIB for Natural Language Processing

- The seven Tarzan books, available from the Gutenberg project (http://promo.net/pg/), were *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 preprocessing as in section 6.1.1, ending up with a sequence of 580,806 words taken from a vocabulary of 19,458 distinct words.
- The 10 most frequent words in the above books that are not stop words (i.e., W values) were *apemans*, *apes*, *eyes*, *girl*, *great*, *jungle*, *tarzan*, *time*, *two*, and *way*.
- After removing triplets with fewer than three occurrences, we had 672 different triplets with a total of 4,479 occurrences. 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 $|\mathcal{W}_p| = 90,\ |\mathcal{W}| = 10,\ |\mathcal{W}_n| = 233$.

- As in the symmetric IB, we first randomly initialize T_p and optimize it using the original sIB algorithm (Slonim et al., 2002), such that $I(T_p; W)$ is maximized. Similarly, we find T_n such that $I(T_n; W)$ is maximized. Using these initializations and the general scheme described in Figure 6, 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 the predictions over the new sequence (*The Son of Tarzan*), for each of the 10 words in W, we define the following quantities: $A_1(w)$ is the number of w occurrences correctly predicted as w (true positives); $A_2(w)$ is the number of words incorrectly predicted as w (false positives); $A_3(w)$ is the number of w occurrences incorrectly not predicted as w (false negatives); The precision and recall for w are 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 microaveraged precision and recall are defined by (Sebastiani, 2002):

$$\begin{cases} < Prec >= \frac{\sum_w A_1(w)}{\sum_w A_1(w)+A_2(w)}, \\ < Rec >= \frac{\sum_w A_1(w)}{\sum_w A_1(w)+A_3(w)}. \end{cases} \tag{B.1}$$

Appendix C: Notations

Capital letters, $\{X, Y, T\}$, denote names of random variables. Lowercase letters, $\{x, y, t\}$, denote specific values taken by these variables. $p(X)$ denotes the probability distribution function, and $p(x)$ denotes the specific (scalar) value, $p(X = x)$—the probability that the assignment to the random variable X is the specific value x. We further use $X \sim p(X)$ to denote that X is distributed according to $p(X)$.

Probability distributions that are given as input and do not change during the analysis are denoted by $p(\cdot)$, while probability distributions that involve changeable parameters are denoted by $q(\cdot)$.

Calligraphic notations, $\{\mathcal{X}, \mathcal{Y}, \mathcal{T}\}$, denote 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, we limit the discussion to discrete random variables with a finite number of possible values.

Sets of random variables are denoted by boldface capital letters $\{\mathbf{X}, \mathbf{T}\}$ and specific values taken by those sets by boldface lowercase letters $\{\mathbf{x}, \mathbf{t}\}$.

The boldface calligraphic notation $\mathcal{T}$ stands for the set of all possible assignments to $\mathbf{T}$.

Acknowledgments

Insightful discussions with Ori Mosenzon are greatly appreciated. We thank Gill Bejerano who prepared the GST proteins data set and brought to our attention the existence of the new *Omega* class in these data. We thank Esther Singer and Michal Rosen-Zvi for comments on previous drafts of this article. This work was supported in part by the Israel Science Foundation (ISF), the Israeli Ministry of Science, and the US-Israel Bi-National Science Foundation. N.S. was also supported by an Eshkol fellowship. N.F. was also supported by an Alon fellowship and the Harry and Abe Sherman Senior Lectureship in Computer Science. Experiments reported in this work were run on equipment funded by an ISF Basic Equipment Grant.

References

Apweiler, R., and Attwood, T. K., Bairoch, A., Bateman, A., Birney, E., Biswas, M., Bucher, P., Cerutti, L., Corpet, F., Croning, M. D., Durbin, R., Falquet, L., Fleischmann, W., Gouzy, J., Hermjakob, H., Hulo, N., Jonassen, I., Kahn, D., Kanapin, A., Karavidopoulou, Y., Lopez, R., Marx, B., Mulder, N. J., Oinn, T. M., Pagni, M., Servant, F., Sigrist, C. J., & Zdobnov, E. M. (2000). InterPro—an integrated documentation resource for protein families, domains and functional sites, *Bioinformatics, 16*, 1145–1150.

Attwood, T. K., Croning, M. D. R., Flower, D. R., Lewis, A. P., Mabey, J. E., Scordis, P., Selley, J., & Wright, W. (2000). PRINTS-S: The database formerly known as PRINTS. *Nucl. Acids Res., 28*, 225–227.

Atwal, G. S., & Slonim, N. (2005). *Information bottleneck with finite samples*. Unpublished manuscript.

Baker, L. D., & McCallum, A. K. (1998). Distributional clustering of words for text classification. In *Proc. of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM.

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Comp., 7*, 1129–1159.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Csiszár, I., & Tusnády, G. (1984). Information geometry and alternating minimization procedures. *Statistics and Decisions*, suppl. 1, 205–237.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B, 39*, 1–38.

Elidan, G., & Friedman, N. (2003). The information bottleneck expectation maximization algorith. In *Proc. of the 19th Conf. on Uncertainty in Artificial Intelligence (UAI-19)*. San Mateo, CA: Morgan Kaufmann.

Friedman, N., Mosenzon, O., Slonim, N., & Tishby, N. (2001). Multivariate information bottleneck. In *Proc. of Uncertainty in Artificial Intelligence (UAI-17)*. San Mateo, CA: Morgan Kaufmann.

Gilad-Bachrach, R., Navot, A., & Tishby, N. (2003). An information theoretic tradeoff between complexity and accuracy. In *The Sixteenth Annual Conference on Learning Theory (COLT)*. New York: Springer.

Globerson, A., & Tishby, N. (2004). The minimum information principle in discriminative learning. In C. Meek, M. Chickering, & J. Halpern (Eds.), *Uncertainty in artificial intelligence*. Banff, Canada: AUAI Press.

Golub, T., Slonim, D., Tamayo, P., Huard, C. M., Caasenbeek, J. M., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., & Lander, E. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science, 286*, 531–537.

Gutman, M. (1989). Asymptotically optimal classification for multiple tests with empirically observed statistics, *IEEE Trans. Inf. Theory, 35*(2), 401–408.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *Proc. of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 50–57). New York: ACM.

Lang, K. (1995). Learning to filter netnews. In *Proc. of the 12th International Conf. on Machine Learning (ICML)*. San Mateo, CA: Morgan Kaufmann.

Parker, E. A., Gedeon, T., & Dimitrov, A. G. (2002). Annealing and the rate distortion problem. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems, 15* (pp. 969–976). Cambridge, MA: MIT Press.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. San Francisco: Morgan Kaufmann.

Pereira, F. C., Tishby, N., & Lee, L. (1993). Distributional clustering of English words. In *30th Annual Meeting of the Association for Computational Linguistics*. New York: ACM.

Rose, K. (1998). Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proceedings of the IEEE, 86*, 2210–2239.

Schriebman, A. (2000). *Stochastic modeling for efficient computation of information theoretic quantities*. Unpublished doctoral dissertation, Hebrew University, Jerusalem, Israel. Available online at http://www.cs.huji.ac.il/labs/learning/Theses/theses_list.html.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1–47.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell Systems Technical Journal, 27*, 379–423, 623–656.

Slonim, N. (2002). *The information bottleneck: Theory and applications*. Unpublished doctoral dissertation, Hebrew University, Jerusalem, Israel.

Slonim, N., Friedman, N., & Tishby, N. (2002). Unsupervised document classification using sequential information maximization. In *Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York: ACM.

Slonim, N., Somerville, R., Tishby, N., & Lahav, O. (2001). Objective classification of galaxy spectra using the information bottleneck method. *Monthly Notes of the Royal Astronomical Society, 323*, 270–284.

Slonim, N., & Tishby, N. (1999). Agglomerative information bottleneck. In S. A. Solla, T. K. Leen, & K.-R. Müller (Eds.), *Advances in neural information processing systems, 12* (pp. 617–623). Cambridge, MA: MIT Press.

Slonim, N., & Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In *Proc. of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 208–215). New York: ACM.

Slonim, N., & Weiss, Y. (2002). Maximum likelihood and the information bottleneck. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems, 15*. Cambridge, MA: MIT Press.

Still, S., & Bialek, W. (2003). How many clusters? *Physics*/030101.

Studeny, M., & Vejnarova, J. (1998). The Multi-information function as a tool for measuring stochastic dependence. In M. I. Jordan (Ed.), *Learning in graphical models* (pp. 261–298). Dordrecht: Kluwer.

Tishby, N., Pereira, F., & Bialek, W. (1999). The information bottleneck method. In *Proc. of 37th Allerton Conference on Communication and Computation.*

Tishby, N., & Slonim, N. (2000). Data clustering by Markovian relaxation and the information bottleneck method. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems, 13*. Cambridge, MA: MIT Press.

Received February 25, 2005; accepted September 15, 2005.