The Deterministic Information Bottleneck

DJ Strouse
danieljstrouse@gmail.com
Department of Physics, Princeton University, Princeton, NJ 08544, U.S.A.

David J. Schwab
david.schwab@northwestern.edu
Department of Physics, Northwestern University, Evanston, IL 60208, U.S.A.

Lossy compression and clustering fundamentally involve a decision about which features are relevant and which are not. The information bottleneck method (IB) by Tishby, Pereira, and Bialek (1999) formalized this notion as an information-theoretic optimization problem and proposed an optimal trade-off between throwing away as many bits as possible and selectively keeping those that are most important. In the IB, compression is measured by mutual information. Here, we introduce an alternative formulation that replaces mutual information with entropy, which we call the deterministic information bottleneck (DIB) and argue better captures this notion of compression. As suggested by its name, the solution to the DIB problem turns out to be a deterministic encoder, or hard clustering, as opposed to the stochastic encoder, or soft clustering, that is optimal under the IB. We compare the IB and DIB on synthetic data, showing that the IB and DIB perform similarly in terms of the IB cost function, but that the DIB significantly outperforms the IB in terms of the DIB cost function. We also empirically find that the DIB offers a considerable gain in computational efficiency over the IB, over a range of convergence parameters. Our derivation of the DIB also suggests a method for continuously interpolating between the soft clustering of the IB and the hard clustering of the DIB.

1 Introduction

Compression is a ubiquitous task for humans and machines alike (Cover & Thomas, 2006; MacKay, 2002). For example, machines must turn the large pixel grids of color that form pictures into small files capable of being shared quickly on the web (Wallace, 1991), humans must compress the vast stream of ongoing sensory information they receive into small changes in the brain that form memories (Kandel, Schwartz, Jessell, Siegelbaum, & Hudspeth, 2013), and data scientists must turn large amounts of high-dimensional

and messy data into more manageable and interpretable clusters (MacKay, 2002).

Lossy compression involves an implicit decision about what is relevant and what is not (Cover & Thomas, 2006; MacKay, 2002). In the example of image compression, the algorithms we use deem some features essential to representing the subject matter well, and others are thrown away. In the example of human memory, our brains deem some details important enough to warrant attention, and others are forgotten. And in the example of data clustering, information about some features is preserved in the mapping from data point to cluster ID, while information about others is discarded.

In many cases, the criterion for "relevance" can be described as information about some other variable(s) of interest. We call X the signal we are compressing, T the compressed version, Y the other variable of interest, and $I(T; Y)$ the "information" that T has about Y (we formally define this later). For human memory, X is past sensory input, T the brain's internal representation (e.g., the activity of a neural population, or the strengths of a set of synapses), and Y the features of the future environment that the brain is interested in predicting, such as extrapolating the position of a moving object. Thus, $I(T; Y)$ represents the predictive power of the memories formed (Palmer, Marre, Berry, & Bialek, 2015). For data clustering, X is the original data, T is the cluster ID, and Y is the target for prediction (e.g., purchasing or ad-clicking behavior in a user segmentation problem). In summary, a good compression algorithm can be described as a trade-off between the compression of a signal and the selective maintenance of the relevant bits that help predict another signal.

This problem was formalized as the information bottleneck (IB) by Tishby, Pereira, and Bialek (1999). Their formulation involved an information-theoretic optimization problem and resulted in an iterative soft clustering algorithm guaranteed to converge to a local (though not necessarily global) optimum. In their cost functional, compression was measured by the mutual information $I(X; T)$. This compression metric has its origins in rate-distortion theory and channel coding, where $I(X; T)$ represents the maximal information transfer rate, or capacity, of the communication channel between X and T (Cover & Thomas, 2006).[1] While this approach has its applications, often one is more interested in directly restricting the amount of resources required to represent T, quantified by the entropy $H(T)$. This latter notion comes from the source coding literature and implies a restriction on the representational cost of T (Cover & Thomas, 2006). In the case of human memory, for example, $H(T)$ would roughly correspond to the

---

[1]Note that the IB problem setting differs significantly from that of channel coding, however. In channel coding, the channel is fixed, and we are free to vary the input distribution, while in IB, the input distribution is fixed, and we are free to vary the channel.

number of neurons or synapses required to represent or store a sensory signal X. In the case of data clustering, $H(T)$ is related to the number of clusters.

In this letter, we introduce an alternative formulation of the IB, the deterministic information bottleneck (DIB), replacing the compression measure $I(X; T)$ with $H(T)$, thus emphasizing contraints on representation rather than communication. Using a clever generalization of both cost functionals, we derive an iterative solution to the DIB, which turns out to provide a hard clustering, or deterministic mapping from X to T, as opposed to the soft clustering, or probabilitic mapping, that IB provides. Finally, we compare the IB and DIB solutions on synthetic data to help illustrate their differences.

2 The Original Information Bottleneck

Given the joint distribution $p(x, y)$, the encoding distribution $q(t|x)$ is obtained through the following information bottleneck (IB) optimization problem:

$$\min_{q(t|x)} L[q(t|x)] = I(X; T) - \beta I(Y; T), \tag{2.1}$$

subject to the Markov constraint $T \leftrightarrow X \leftrightarrow Y$. Here $I(X; T)$ denotes the mutual information between X and T, that is, $I(X; T) \equiv H(T) - H(T|X) = \sum_{x,t} p(x, t) \log\left(\frac{p(x,t)}{p(x)p(t)}\right) = D_{KL}\big[p(x, t) \mid p(x)\, p(t)\big]$,[2] where D_{KL} denotes the Kullback-Leibler divergence.[3] The first term in the cost function is meant to encourage compression and the second, relevance. β is a nonnegative free parameter representing the relative importance of compression and relevance, and our solution will be a function of it. The Markov constraint

---

[2]Implicit in the summation here, we have assumed that X, Y, and T are discrete. We will be keeping this assumption throughout for convenience of notation, but note that the IB generalizes naturally to X, Y, and T continuously by simply replacing the sums with integrals (see, e.g., Chechik, Globerson, Tishby, & Weiss, 2005).

[3]For those unfamiliar with it, mutual information is a very general measure of how related two variables are. Classic correlation measures typically assume a certain form of the relationship between two variables, say, linear, whereas mutual information is agnostic as to the details of the relationship. One intuitive picture comes from the entropy decomposition: $I(X; Y) \equiv H(X) - H(X|Y)$. Since entropy measures uncertainty, mutual information measures the reduction in uncertainty in one variable when observing the other. Moreover, it is symmetric ($I(X; Y) = I(Y; X)$), so the information is mutual. Another intuitive picture comes from the D_{KL} form: $I(X; Y) \equiv D_{KL}\big[p(x, y) \mid p(x)\, p(y)\big]$. Since D_{KL} measures the distance between two probability distributions, the mutual information quantifies how far the relationship between x and y is from a probabilistically independent one, that is, how far the joint $p(x, y)$ is from the factorized $p(x)\, p(y)$. A very nice summary of mutual information as a dependence measure is included in Kinney and Atwal (2014).

simply enforces the probabilistic graphical structure of the task; the compressed representation T is a (possibly stochastic) function of X and can get information about Y only through X. Note that we are using p to denote distributions that are given and fixed, and q to denote distributions that we are free to change and are being updated throughout the optimization process.

Through a standard application of variational calculus (see section 8 for a detailed derivation of the solution to a more general problem introduced below), one finds the formal solution[4]

$$q(t|x) = \frac{q(t)}{Z(x, \beta)} \exp\left[-\beta D_{\mathrm{KL}}\left[p(y|x) \mid q(y|t)\right]\right], \tag{2.2}$$

$$q(y|t) = \frac{1}{q(t)} \sum_x q(t|x) p(x, y), \tag{2.3}$$

where $Z(x, \beta) \equiv \exp\left[-\frac{\lambda(x)}{p(x)} - \beta \sum_y p(y \mid x) \log \frac{p(y|x)}{p(y)}\right]$ is a normalization factor and $\lambda(x)$ is a Lagrange multiplier that enters enforcing normalization of $q(t \mid x)$.[5] To get an intuition for this solution, it is useful to take a clustering perspective. Since we are compressing X into T, many X will be mapped to the same T, and so we can think of the IB as clustering xs into their cluster labels t. The solution $q(t|x)$ is then likely to map x to t when $D_{\mathrm{KL}}\left[p(y|x) \mid q(y|t)\right]$ is small or, in other words, when the distributions $p(y|x)$ and $q(y|t)$ are similar. These distributions are similar to the extent that x and t provide similar information about y. In summary, inputs x get mapped to clusters t that maintain information about y, as was desired.

This solution is formal because the first equation depends on the second and vice versa. However, Tishby et al. (1999) showed that an iterative approach can be built on the the above equations, which provably converges to a local optimum of the IB cost function (see equation 2.1).

Starting with some initial distributions $q^{(0)}(t|x)$, $q^{(0)}(t)$, and $q^{(0)}(y|t)$, the nth update is given by[6]

---

[4]For readers familiar with rate-distortion theory, equation 2.2 can be viewed as the solution to a rate-distortion problem with the distortion measure given by the KL divergence term in the exponent.

[5]More explicitly, our cost function L also implicitly includes a term $\sum_x \lambda(x)\left[1 - \sum_t q(t|x)\right]$, and this is where $\lambda(x)$ comes into the equation. See section 8 for details.

[6]Note that if at step m, no xs are assigned to a particular $t = t^*$ (i.e., $q(t \mid x) = 0\ \forall x$), then $q^{(m)}(t^*) = q^{(m+1)}(t^*) = 0$. That is, no xs will ever again be assigned to t^* (due to the $q^{(n-1)}(t)$ factor in $q^{(n-1)}(t \mid x)$). In other words, the number of ts "in use" can only decrease during the iterative algorithm (or remain constant). Thus, it seems plausible that the solution will not depend on the cardinality of T, provided it is chosen to be large enough.

Algorithm 1: The Information Bottleneck Method.

1: Given $p(x, y)$, $\beta \geq 0$
2: Initialize $q^{(0)}(t \mid x)$
3: $q^{(0)}(t) = \sum_x p(x) \, q^{(0)}(t \mid x)$
4: $q^{(0)}(y \mid t) = \frac{1}{q^{(0)}(t)} \sum_x p(x, y) \, q^{(0)}(t \mid x)$
5: $n = 0$
6: **while** not converged **do**
7:      $n = n + 1$
8:      $q^{(n)}(t \mid x) = \frac{q^{(n-1)}(t)}{Z(x, \beta)} \exp\left[-\beta D_{\mathrm{KL}}\left[p(y \mid x) \mid q^{(n-1)}(y \mid t)\right]\right]$
9:      $q^{(n)}(t) = \sum_x p(x) \, q^{(n)}(t \mid x)$
10:     $q^{(n)}(y \mid t) = \frac{1}{q^{(n)}(t)} \sum_x q^{(n)}(t \mid x) \, p(x, y)$
11: **end while**

$$q^{(n)}(t|x) = \frac{q^{(n-1)}(t)}{Z^{(n)}(x, \beta)} \exp[-\beta D_{\mathrm{KL}}[p(y|x) \mid q^{(n-1)}(y|t)]], \qquad (2.4)$$

$$q^{(n)}(t) = \sum_x q^{(n)}(t|x) p(x), \qquad (2.5)$$

$$q^{(n)}(y|t) = \frac{1}{q^{(n)}(t)} \sum_x q^{(n)}(t|x) p(x, y). \qquad (2.6)$$

Note that the first equation, 2.4, is the only "meaty" one; the other two, 2.5 and 2.6, are just there to enforce consistency with the laws of probability (e.g., that marginals are related to joints as they should be). In principle, with no proof of convergence to a global optimum, it might be possible for the solution obtained to vary with the initialization, but in practice, the cost function is smooth enough that this does not seem to happen. This algorithm is summarized in algorithm 1. Note that while the general solution is iterative, there is at least one known case in which an analytic solution is possible: when X and Y are jointly gaussian (Chechik, Globerson, Tishby, & Weiss, 2005).

In summary, given the joint distribution $p(x, y)$, the IB method extracts a compressive encoder $q(t \mid x)$ that selectively maintains the bits from X that are informative about Y. As the encoder is a function of the free parameter β, we can visualize the entire family of solutions on a curve (see Figure 1), showing the trade-off between compression (on the x-axis) and relevance (on the y-axis), with β as an implicitly varying parameter. For small β, compression is more important than prediction, and we find ourselves at the bottom left of the curve in the high-compression, low-prediction regime. As β increases, prediction becomes more important relative to compression, and we see that both $I(X; T)$ and $I(T; Y)$ increase. At some point, $I(T; Y)$ saturates, because there is no more information about Y that can be extracted
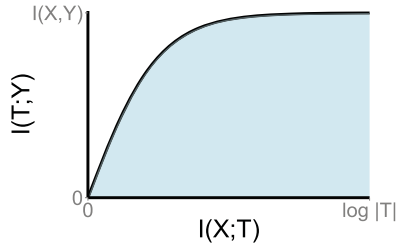
Figure 1: An illustrative IB curve. $I(T; Y)$ is the relevance term from equation 2.1; $I(X; T)$ is the compression term. $I(X; Y)$ is an upper bound on $I(T; Y)$ since T gets its information about Y only via X. $\log(|T|)$, where $|T|$ is the cardinality of the compression variable, is a bound on $I(X; T)$ since $I(X; T) = H(T) - H(T \mid X) \leq H(T) \leq \log(|T|)$.

from X (either because $I(T; Y)$ has reached $I(X; Y)$ or because T has too small cardinality). In this regime, the encoder will approach the trivially deterministic solution of mapping each x to its own cluster. At any point on the curve, the slope is equal to β^{-1}, which we can read off directly from the cost functional. Note that the region below the curve is shaded because this area is feasible; for suboptimal $q(t \mid x)$, solutions will lie in this region. Optimal solutions will, of course, lie on the curve, and no solutions will lie above the curve.

Additional work on the IB has highlighted its relationship with maximum likelihood on a multinomial mixture model (Slonim & Weiss, 2002) and canonical correlation analysis (Creutzig, Globerson, & Tishby, 2009)—and therefore linear gaussian models (Bach & Jordan, 2006) and slow feature analysis (Turner & Sahani, 2007). Applications have included speech recognition (Hecht & Tishby, 2005; Hecht, Noor, & Tishby, 2009), topic modeling (Slonim & Tishby, 2000b, 2001; Bekkerman, El-Yaniv, Tishby, & Winter, 2001, 2003), and neural coding (Schneidman, Slonim, Tishby, de Ruyter van Steveninck, & Bialek, 2001; Palmer, Marre, Berry, & Bialek, 2015). Most recently, the IB has even been proposed as a method for benchmarking the performance of deep neural networks (Tishby & Zaslavsky, 2015).

3 The Deterministic Information Bottleneck

Our motivation for introducing an alternative formulation of the information bottleneck is rooted in the compression term of the IB cost function; there, the minimization of the mutual information $I(X; T)$ represents compression. As discussed above, this measure of compression comes from the channel coding literature and implies a restriction on the communication cost between X and T. Here, we are interested in the source coding notion of compression, which implies a restriction on the representational cost of T.

For example, in neuroscience, there is a long history of work on redundancy reduction in the brain in the form of minimizing $H(T)$ (Barlow, 1981, 2001a, 2001b).

Let us call the original IB cost function L_{IB}, that is, $L_{IB} \equiv I(X;T) - \beta I(T;Y)$. We now introduce the deterministic information bottleneck (DIB) cost function,

$$L_{DIB}[q(t|x)] \equiv H(T) - \beta I(T;Y), \qquad (3.1)$$

which is to be minimized over $q(t \mid x)$, and subject to the same Markov constraint as the original formulation, equation 2.1. The motivation for the "deterministic" in its name will become clear in a moment.

To see the distinction between the two cost functions, note that

$$L_{IB} - L_{DIB} = I(X;T) - H(T) \qquad (3.2)$$
$$= -H(T \mid X), \qquad (3.3)$$

where we have used the decomposition of the mutual information $I(X;T) = H(T) - H(T \mid X)$. $H(T \mid X)$ is sometimes called the noise entropy and measures the stochasticity in the mapping from X to T. Since we are minimizing these cost functions, this means that the IB cost function encourages stochasticity in the encoding distribution $q(t \mid x)$ relative to the DIB cost function. In fact, we will see that by removing this encouragement of stochasticity, the DIB problem actually produces a deterministic encoding distribution, that is, an encoding function, hence, the "deterministic" in its name.

Naively taking the same variational calculus approach as for the IB problem, one cannot solve the DIB problem.[7] To make this problem tractable, we are going to define a family of cost functions of which the IB and DIB cost functions are limiting cases. That family, indexed by α, is defined as[8]

$$L_\alpha \equiv H(T) - \alpha H(T \mid X) - \beta I(T;Y). \qquad (3.4)$$

---

[7]When you take the variational derivative of the L_{DIB} + Lagrange multiplier term with respect to $q(t \mid x)$ and set it to zero, you get no explicit $q(t \mid x)$ term, and it is therefore not obvious how to solve these equations. We cannot rule out that approach is possible, of course; we have just taken a different route here.

[8]Note that for $\alpha < 1$, we cannot allow T to be continuous since $H(T)$ can become infinitely negative, and the optimal solution in that case will trivially be a delta function over a single value of T for all X, across all values of β. This is in contrast to the IB, which can handle continuous T. In any case, we continue to assume discrete X, Y, and T for convenience.

Clearly, $L_{IB} = L_1$. However, instead of looking at L_{DIB} as the $\alpha = 0$ case, we define the DIB solution $q_{DIB}(t \mid x)$ as the $\alpha \to 0$ limit of the solution to the generalized problem $q_\alpha(t \mid x)$:[9]

$$q_{DIB}(t \mid x) \equiv \lim_{\alpha \to 0} q_\alpha(t \mid x). \tag{3.5}$$

Taking the variational calculus approach to minimizing L_α (under the Markov constraint), we get the following solution for the encoding distribution (see the appendix for the derivation and explicit form of the normalization factor $Z(x, a, \beta)$):

$$q_\alpha(t|x) = \frac{1}{Z(x, \alpha, \beta)} \exp\left[\frac{1}{\alpha}\left(\log q_\alpha(t) - \beta D_{KL}[p(y|x) \mid q_\alpha(y|t)]\right)\right], \tag{3.6}$$

$$q_\alpha(y|t) = \frac{1}{q_\alpha(t)} \sum_x p(y|x)\, q_\alpha(t|x) p(x). \tag{3.7}$$

Note that the last equation is just equation 2.3, since this follows from the Markov constraint. With $\alpha = 1$, we can see that the first equation becomes the IB solution from equation 2.2, as should be the case.

Before we take the $\alpha \to 0$ limit, note that we can now write a generalized IB iterative algorithm (indexed by α) that includes the original as a special case ($\alpha = 1$):

$$q_\alpha^{(n)}(t|x) = \frac{1}{Z(x, \alpha, \beta)} \exp\left[\frac{1}{\alpha}(\log q_\alpha^{(n-1)}(t) - \beta D_{KL}[p(y|x) \mid q_\alpha^{(n-1)}(y|t)])\right], \tag{3.8}$$

$$q_\alpha^{(n)}(t) = \sum_x p(x)\, q_\alpha^{(n)}(t|x), \tag{3.9}$$

$$q_\alpha^{(n)}(y|t) = \frac{1}{q_\alpha^{(n)}(t)} \sum_x q_\alpha^{(n)}(t|x) p(x, y). \tag{3.10}$$

This generalized algorithm can be used in its own right; however, we will not discuss that option further here.

For now, we take the limit $\alpha \to 0$ and see that something interesting happens with $q_\alpha(t \mid x)$: the argument of the exponential begins to blow up. For a fixed x, this means that $q(t \mid x)$ will collapse into a delta function at the value of t which maximizes $\log q(t) - \beta D_{KL}[p(y \mid x) \mid q(y \mid t)]$, that is,

---

[9]Note a subtlety here that we cannot claim that the q_{DIB} is the solution to L_{DIB}, for although $L_{DIB} = \lim_{\alpha \to 0} L_\alpha$ and $q_{DIB} = \lim_{\alpha \to 0} q_\alpha$, the solution of the limit need not be equal to the limit of the solution. It would, however, be surprising if that were not the case.

$$\lim_{\alpha \to 0} q_\alpha(t|x) = f : X \to T, \tag{3.11}$$

where

$$f(x) = t^* = \underset{t}{\operatorname{argmax}} \left(\log q(t) - \beta D_{\mathrm{KL}}[p(y|x) \mid q(y|t)] \right). \tag{3.12}$$

So, as anticipated, the solution to the DIB problem is a deterministic encoding distribution. The $\log q(t)$ above encourages that we use as few values of t as possible, via a "rich-get-richer" scheme that assigns each x preferentially to a t with many xs already assigned to it. The KL divergence term, as in the original IB problem, just makes sure we pick ts which retain as much information from x about y as possible. The parameter β, as in the original problem, controls the trade-off between how much we value compression and prediction.

Also as in the original problem, the solution above is only a formal solution, since equation 3.6 depends on equation 3.7 and vice versa. We will again need to take an iterative approach; in analogy to the IB case, we repeat the following updates to convergence (from some initialization):[10]

$$f^{(n)}(x) = \underset{t}{\operatorname{argmax}} (\log q^{(n-1)}(t) - \beta D_{\mathrm{KL}}[p(y|x) \mid q^{(n-1)}(y|t)]), \tag{3.13}$$

$$q^{(n)}(t|x) = \delta(t - f^{(n)}(x)), \tag{3.14}$$

$$q^{(n)}(t) = \sum_x q^{(n)}(t|x) p(x), \tag{3.15}$$

$$= \sum_{x: f^{(n)}(x)=t} p(x) \tag{3.16}$$

$$q^{(n)}(y|t) = \frac{1}{q^{(n)}(t)} \sum_x q^{(n)}(t \mid x) \, p(x,y), \tag{3.17}$$

$$= \frac{\sum_{x: f^{(n)}(x)=t} p(x,y)}{\sum_{x: f^{(n)}(x)=t} p(x)}. \tag{3.18}$$

This process is summarized in algorithm 2.

Note that the DIB algorithm also corresponds to "clamping" IB at every step by assigning each x to its highest-probability cluster t. We can see this

---

[10] As with the IB, the DIB has the property that once a cluster goes unused, it will not be brought back into use in future steps. That is, if $q^{(m)}(t) = 0$, then $\log q^{(m)}(t) = -\infty$ and $q^{(m+1)}(t \mid x) = 0 \; \forall x$. So once again, one should conservatively choose the cardinality of T to be "large enough"; for both the IB and DIB, we chose to set it equal to the cardinality of X.

Algorithm 2: The Deterministic Information Bottleneck Method.

1: Given $p(x, y)$, $\beta \geq 0$
2: Initialize $f^{(0)}(x)$
3: Set $q^{(0)}(t) = \sum_{x:f^{(0)}(x)=t} p(x)$
4: Set $q^{(0)}(y \mid t) = \frac{\sum_{x:f^{(0)}(x)=t} p(x,y)}{\sum_{x:f^{(0)}(x)=t} p(x)}$
5: $n = 0$
6: **while** not converged **do**
7:     $n = n + 1$
8:     $d^{(n-1)}(x, t) \equiv D_{\mathrm{KL}}\big[p(y \mid x) \mid q^{(n-1)}(y \mid t)\big]$
9:     $\ell_{\beta}^{(n-1)}(x, t) \equiv \log q(t) - \beta d^{(n-1)}(x, t)$
10:     $f^{(n)}(x) = \underset{t}{\mathrm{argmax}} \, \ell_{\beta}^{(n-1)}(x, t)$
11:     $q^{(n)}(t) = \sum_{x:f^{(n)}(x)=t} p(x)$
12:     $q^{(n)}(y \mid t) = \frac{\sum_{x:f^{(n)}(x)=t} p(x,y)}{\sum_{x:f^{(n)}(x)=t} p(x)}$
13: **end while**

---

by taking the argmax of the logarithm of $q(t \mid x)$ in equation 2.2, noting that the argmax of a positive function is equivalent to the argmax of its logarithm, discarding the $\log(Z(x, \beta))$ term since it does not depend on t, and seeing that the result corresponds to equation 3.12. We emphasize, however, that this is not the same as simply running the IB algorithm to convergence and then clamping the resulting encoder; that would, in most cases, produce a suboptimal, "unconverged" deterministic solution.

As with the IB, the DIB solutions can be plotted as a function of β. However, in this case, it is more natural to plot $I(T; Y)$ as a function of $H(T)$ rather than $I(X; T)$. That said, in order to compare the IB and DIB, they need to be plotted in the same plane. When plotting in the $I(X; T)$ plane, the DIB curve will of course lie below the IB curve, since in this plane, the IB curve is optimal; the opposite will be true when plotting in the $H(T)$ plane. Comparisons with experimental data can be performed in either plane. A Python implementation of algorithms 1 and 2, as well as tools for generating synthetic data and the analysis of results, is available at https://github.com/djstrouse/information-bottleneck.

4 Comparison of IB and DIB

To get an idea of how the IB and DIB solutions differ in practice, we generated a series of random joint distributions $p(x, y)$, solved for the IB and DIB solutions for each, and compared them in both the IB and DIB plane. To generate the $p(x, y)$, we first sampled $p(x)$ from a symmetric Dirichlet distribution with concentration parameter α_x (so $p(x) \sim \mathrm{Dir}[\alpha_x]$) and then
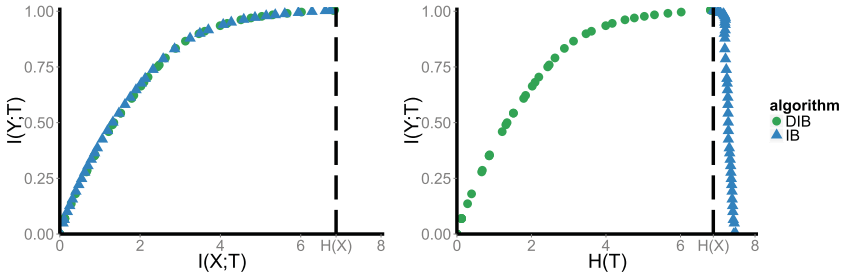
Figure 2: Example IB and DIB solutions. (Left) IB plane. (Right) DIB plane. The upper limit of the y-axes is $I(X, Y)$, since this is the maximal possible value of $I(T; Y)$. The upper limit of the x-axes is $\log(|T|)$, since this is the maximal possible value of $H(T)$ and $I(X, T)$ (the latter being true since $I(X, T)$ is bounded above by both $H(T)$ and $H(X)$, and $|T| < |X|$). The dashed vertical lines mark $H(X)$, which is both an upper bound for $I(X, T)$ and a natural comparison for $H(T)$ (since to place each data point in its own cluster, we need $H(T) = H(X)$).

sampled each row of $p(y \mid x)$ from another symmetric Dirichlet distribution with concentration parameter $\alpha_y^{(i)}$ (so $p(y \mid x_i) \sim \text{Dir}[\alpha_y^{(i)}]$). In the experiments shown here, we set α_x to 1000, so that each x_i was approximately equally likely, and we set $\alpha_y^{(i)}$ to be equally spaced logarithmically between $10^{-1.3}$ and $10^{1.3}$ in order to provide a range of informativeness in the conditionals. We set the cardinalities of X and Y to $|X| = 256$ and $|Y| = 32$, with $|X| > |Y|$ for two reasons. First, this encourages overlap between the conditionals $p(y|x)$, which leads to a more interesting clustering problem. Second, in typical applications, this will be the case, such as in document clustering, where there are often many more documents than vocabulary words. Since the number of clusters in use for both IB and DIB can only decrease from iteration to iteration (see footnote 10), we always initialized $|T| = |X|$.[11] For the DIB, we initialized the cluster assignments to be as even across the cluster as possible; that is, each data point belonged to its own cluster. For IB, we initialized using a soft version of the same procedure, with 75% of each conditional's probability mass assigned to a unique cluster and the remaining 25% a normalized uniform random vector over the remaining $|T| - 1$ clusters.

An illustrative pair of solutions is shown in Figure 2. The key feature to note is that while performance of the IB and DIB solutions is very similar in the IB plane, their behavior differs drastically in the DIB plane.

Perhaps most unintuitive is the behavior of the IB solution in the DIB plane, where from an entropy perspective, the IB actually "decompresses"

---

[11] An even more efficient setting might be to set the cardinality of T based on the entropy of X, say, $|T| = \text{ceiling}(\exp(H(X)))$, but we did not experiment with this.
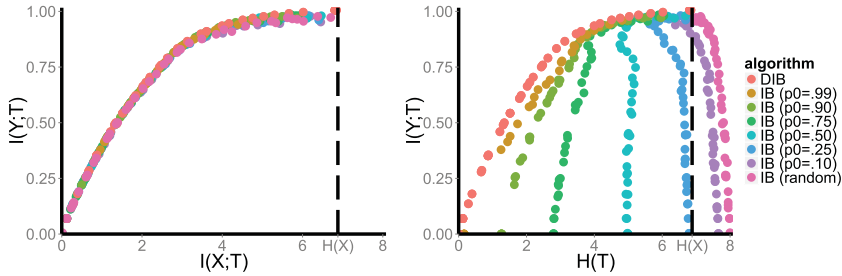
Figure 3: Example IB and DIB solutions across different IB initializations. Details identical to Figure 2, except colors represent different initializations for the IB, as described in the text.

the data (i.e., $H(T) > H(X)$). To understand this behavior, recall that the IB's compression term is the mutual information $I(X, T) = H(T) - H(T \mid X)$. This term is minimized by any $q(t \mid x)$ that maps ts independent of xs. Consider two extremes of such mappings. One is to map all values of X to a single value of T; this leads to $H(T) = H(T \mid X) = I(X, T) = 0$. The other is to map each value of X uniformly across all values of T; this leads to $H(T) = H(T \mid X) = \log |T|$ and $I(X, T) = 0$. In our initial studies, the IB consistently took the latter approach.[12] Since the DIB cost function favors the former approach (and indeed the DIB solution follows this approach), the IB consistently performs poorly by the DIB's standards. This difference is especially apparent at small β, where the compression term matters most, and as β increases, the DIB and IB solutions converge in the DIB plane.

To encourage the IB to perform closer to DIB optimality at small β, we next altered our initialization scheme of $q(t \mid x)$ to one biased toward single-cluster solutions; instead of each x_i having most of its probability mass on a unique cluster t_i, we placed most of the probability mass for each x_i on the same cluster t^*. The intended effect was to start the IB closer to solutions in which all data points were mapped to a single cluster. Results are shown in Figure 3. Here, p_0 is the amount of probability mass placed on the cluster t^*, that is, $q(t^* \mid x) = p_0$, $\forall x$; the probability mass for the remaining $|T| - 1$ clusters was again initialized to a normalized uniform random vector. "Random" refers to an initialization that skips placing the p_0 mass and just initializes each $q(t \mid x_i)$ to a normalized uniform random vector.

We note several features. First, although we can see a gradual shift of the IB toward DIB-like behavior in the DIB plane as $p_0 \to 1$, the IB solutions never quite reach the performance of DIB. Moreover, as $p_0 \to 1$, the single-cluster initializations exhibit a phase transition in which, regardless of β, they "skip" over a sizable fraction of lower-$I(Y; T)$ solutions that

---

[12]Intuitively, this approach is more random and is therefore easier to stumble on during optimization.
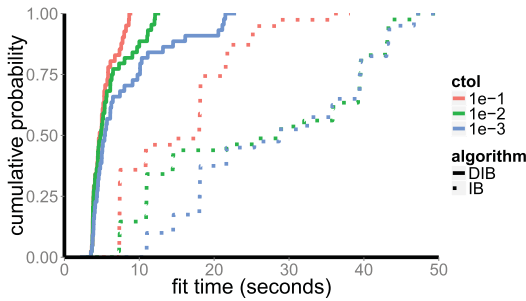
Figure 4: Fit times for IB and DIB. Cumulative distribution function of fit times across β for a variety of settings of the convergence tolerance. Note that absolute numbers here depend on hardware, so we emphasize only relative comparisons of IB versus DIB. Note also that across the range of ctol values we tested here, the (D)IB curves vary by less than the width of the data points, so we omit them.

are picked up by DIB. Third, even for higher-$I(Y; T)$ solutions, the single-cluster initializations seem to perform suboptimally, not quite extracting all of the information $I(X; Y)$, as DIB and the multicluster initialization from the previous section do. This can be seen in both the IB and DIB plane.

To summarize, the IB and DIB perform similarly by the IB standards, but the DIB tends to outperform the IB dramatically by the DIB's standards. Careful initialization of the IB can make up some of the difference, but not all.

It is also worth noting that across all the data sets we tested, the DIB also tended to converge faster, as illustrated in Figure 4. The DIB speedup over IB varied depending on the convergence conditions. In our experiments, we defined convergence as when the relative step-to-step change in the cost functional L was smaller than some threshold ctol, that is, when $\left| \frac{L_{n-1} - L_n}{L_{n-1}} \right| <$ ctol at step n. In the results above, we used ctol $= 10^{-3}$. In Figure 4, we vary ctol, with the IB initialization scheme fixed to the original multicluster version, to show the effect on the relative speedup of DIB over IB. While DIB remained approximately two to five times faster than IB in all cases tested, that speedup tended to be more pronounced with lower ctol. Since the ideal convergence conditions would probably vary by data set size and complexity, it is difficult to make any general conclusions, though our experiments do at least suggest that DIB offers a computational advantage over IB.

5 Related Work

The DIB is not the first hard clustering version of IB.[13] Indeed, the agglomerative information bottleneck (AIB) (Slonim & Tishby, 2000a) also produces a

---

[13]In fact, even the IB itself produces a hard clustering in the large β limit. However, it trivially assigns all data points to their own clusters.

hard clustering and was introduced soon after the IB. Thus, it is important to distinguish between the two approaches. AIB is a bottom-up, greedy method that starts with all data points belonging to their own clusters and iteratively merges clusters in a way that maximizes the gain in relevant information. It was explicitly designed to produce a hard clustering. DIB is a top-down method derived from a cost function that was not designed to produce a hard clustering. Our starting point was to alter the IB cost function to match the source coding notion of compression. The emergence of hard clustering in DIB is itself a result. Thus, while AIB does provide a hard clustering version of IB, DIB contributes the following in addition. First, our study emphasizes why a stochastic encoder is optimal for IB, namely, due to the noise entropy term. Second, our study provides a principled, top-down derivation of a hard clustering version of IB, based on an intuitive change to the cost function. Third, our nontrivial derivation also provides a cost function and solution that interpolates between DIB and IB by adding back the noise entropy continuously, that is, with $0 < \alpha < 1$. This interpolation may be viewed as adding a regularization term to DIB, one that may perhaps be useful in dealing with finitely sampled data. Another interpretation of the cost function with intermediate α is as a penalty on both the mutual information between X and T and the entropy of the compression, $H(T)$.

The original IB also provides a deterministic encoding upon taking the limit $\beta \to \infty$ that corresponds to the causal-state partition of histories (Still, Crutchfield, & Ellison, 2010). However, this is the limit of no compression, whereas our approach allows for an entire family of deterministic encoders with varying degrees of compression.

6 Discussion

Here we have introduced the deterministic information bottleneck (DIB) as an alternative to the information bottleneck (IB) for compression and clustering. We have argued that the DIB cost function better embodies the goal of lossy compression of relevant information and shown that it leads to a nontrivial deterministic version of the IB. We have compared the DIB and IB solutions on synthetic data and found that in our experiments, the DIB performs nearly identically to the IB in terms of the IB cost function, but it is far superior in terms of its own cost function. We also noted that the DIB achieved this performance at a computational efficiency two to five times better than the IB.

Of course, in addition to the studies with synthetic data here, it is important to compare the DIB and IB on real-world data sets as well to see whether the DIB's apparent advantages hold—for example, with data sets that have a more explicit hierarchical structure for both algorthms to exploit, such as in topic modeling (Blei, Griffiths, Jordan, & Tenenbaum, 2003; Slonim & Weiss, 2002).

One particular application of interest is maximally informative clustering, where it would be interesting to know how IB and DIB relate to classic clustering algorithms such as k-means (Strouse & Schwab, 2017). Previous work has, for example, offered a principled way of choosing the number of clusters based on the finiteness of the data (Still & Bialek, 2004), and similarly interesting results may exist for the DIB. More generally, there are learning theory results showing generalization bounds on IB for which an analog on DIB would be interesting as well (Shamir, Sabato, & Tishby, 2010).

Another potential area of application is modeling the extraction of predictive information in the brain (which is one particular example in a long line of work on the exploitation of environmental statistics by the brain (Barlow, 1981, 2001a, 2001b; Atick & Redlich, 1992; Olshausen & Field, 1996, 1997, 2004; Simoncelli & Olshausen, 2001)). There, X would be the stimulus at time t, Y the stimulus a short time in the future $t + \tau$, and T the activity of a population of sensory neurons. One could even consider neurons deeper in the brain by allowing X and Y to correspond not to an external stimulus but to the activity of upstream neurons. An analysis of this nature using retinal data was recently performed with the IB (Palmer et al., 2015). It would be interesting to see if the same data correspond better to the behavior of the DIB, particularly in the DIB plane, where the IB and DIB differ dramatically.

We close by noting that DIB is an imperfect name for the algorithm introduced here for a couple of reasons. First, there exist other deterministic limits and approximations to the IB (see, e.g., the discussion of the AIB in section 5), and so we hesitate to use the phrase "the" deterministic IB. Second, our motivation here was not to create a deterministic version of IB but instead to alter the cost function in a way that better encapsulates the goals of certain problems in data analysis. Thus, the deterministic nature of the solution was a result, not a goal. For this reason, "entropic bottleneck" might also be an appropriate name.

Appendix: Derivation of Generalized IB Solution

Given $p(x, y)$ and subject to the Markov constraint $T \leftrightarrow X \leftrightarrow Y$, the generalized IB problem is

$$\min_{q(t|x)} L[q(t|x)] = H(T) - \alpha H(T|X) - \beta I(Y; T) - \sum_{x,t} \lambda(x)\, q(t|x), \quad \text{(A.1)}$$

where we have now included the Lagrange multiplier term (which enforces normalization of $q(t|x)$) explicitly. The Markov constraint implies the

following factorizations,

$$q(t|y) = \sum_x q(t|x)p(x|y), \tag{A.2}$$

$$q(t) = \sum_x q(t|x)p(x), \tag{A.3}$$

which give us the following useful derivatives:

$$\frac{\delta q(t|y)}{\delta q(t|x)} = p(x|y), \tag{A.4}$$

$$\frac{\delta q(t)}{\delta q(t|x)} = p(x). \tag{A.5}$$

Now, taking the derivative of the cost function with respect to the encoding distribution, we get:

$$\frac{\delta L}{\delta q(t|x)} = -\frac{\delta}{\delta q(t|x)} \sum_t q(t)\log q(t) - \frac{\delta}{\delta q(t|x)} \sum_{x,t} \lambda(x)\,q(t|x) \tag{A.6}$$

$$+\alpha \frac{\delta}{\delta q(t|x)} \sum_{x,t} q(t|x)p(x)\log q(t|x) \tag{A.7}$$

$$-\beta \frac{\delta}{\delta q(t|x)} \sum_{y,t} q(t|y)\,p(y)\log\left[\frac{q(t|y)}{q(t)}\right] \tag{A.8}$$

$$= -\log q(t)\frac{\delta q(t)}{\delta q(t|x)} - q(t)\frac{\delta \log q(t)}{\delta q(t|x)} - \lambda(x)\frac{\delta q(t|x)}{\delta q(t|x)} \tag{A.9}$$

$$+\alpha\left[p(x)\log q(t|x)\frac{\delta q(t|x)}{\delta q(t|x)} + q(t|x)p(x)\frac{\delta \log q(t|x)}{\delta q(t|x)}\right] \tag{A.10}$$

$$-\beta \sum_y\left[p(y)\log\left[\frac{q(t|y)}{q(t)}\right]\frac{\delta q(t|y)}{\delta q(t|x)}\right] \tag{A.11}$$

$$+\beta \sum_y\left[q(t|y)\,p(y)\frac{\delta \log q(t|y)}{\delta q(t|x)} + q(t|y)\,p(y)\frac{\delta \log q(t)}{\delta q(t|x)}\right] \tag{A.12}$$

$$= -p(x)\log q(t) - p(x) - \lambda(x) + \alpha\left[p(x)\log q(t|x) + p(x)\right] \tag{A.13}$$

$$-\beta \sum_y\left[p(y)\log\left[\frac{q(t|y)}{q(t)}\right]p(x|y) + p(y)\,p(x|y) - q(t|y)\,p(y)\frac{p(x)}{q(t)}\right] \tag{A.14}$$

$$= -p(x)\log q(t) - p(x) - \lambda(x) + \alpha\left[p(x)\log q(t|x) + p(x)\right] \quad (A.15)$$

$$-\beta p(x)\left[\sum_y p(y|x)\log\left[\frac{q(t|y)}{q(t)}\right] + \sum_y p(y|x) - \sum_y q(y|t)\right] \quad (A.16)$$

$$= p(x)\left[-1 - \log q(t) - \frac{\lambda(x)}{p(x)} + \alpha\log q(t|x) + \alpha\right.$$

$$\left. -\beta\left[\sum_y p(y|x)\log\left[\frac{q(t|y)}{q(t)}\right]\right]\right]. \quad (A.17)$$

Setting this to zero implies that

$$\alpha\log q(t|x) = 1 - \alpha + \log q(t) + \frac{\lambda(x)}{p(x)} + \beta\left[\sum_y p(y|x)\log\left[\frac{q(t|y)}{q(t)}\right]\right]. \quad (A.18)$$

We want to rewrite the β term as a KL divergence. First, we need that $\log\left[\frac{q(t|y)}{q(t)}\right] = \log\left[\frac{q(t,y)}{q(t)p(y)}\right] = \log\left[\frac{q(y|t)}{p(y)}\right]$. Second, we add and subtract $\beta\sum_y p(y|x)\log\left[\frac{p(y|x)}{p(y)}\right]$. This gives us

$$\alpha\log q(t|x) = 1 - \alpha + \log q(t) + \frac{\lambda(x)}{p(x)} + \beta\sum_y p(y|x)\log\left[\frac{p(y|x)}{p(y)}\right] \quad (A.19)$$

$$-\beta\left[\sum_y p(y|x)\log\left[\frac{p(y|x)}{q(y|t)}\right]\right]. \quad (A.20)$$

The second β term is now just $D_{\mathrm{KL}}[p(y|x)\,|\,q(y|t)]$. Dividing both sides by α, this leaves us with

$$\log q(t|x) = z(x,\alpha,\beta) + \frac{1}{\alpha}\log q(t) - \frac{\beta}{\alpha}D_{\mathrm{KL}}[p(y\,|\,x)\,|q(y|t)], \quad (A.21)$$

where we have absorbed all of the terms that do not depend on t into a single factor:

$$z(x, \alpha, \beta) \equiv \frac{1}{\alpha} - 1 + \frac{\lambda(x)}{\alpha p(x)} + \frac{\beta}{\alpha} \sum_y p(y \mid x) \log \left[\frac{p(y \mid x)}{p(y)} \right]. \quad \text{(A.22)}$$

Solving for $q(t|x)$, we get

$$q(t|x) = \exp[z] \exp \left[\frac{1}{\alpha} \left(\log q(t) - \beta D_{\mathrm{KL}} \big[p(y|x) \mid q(y|t) \big] \right) \right] \quad \text{(A.23)}$$

$$= \frac{1}{Z} \exp \left[\frac{1}{\alpha} \left(\log q(t) - \beta D_{\mathrm{KL}} \big[p(y|x) \mid q(y|t) \big] \right) \right], \quad \text{(A.24)}$$

where

$$Z(x, \alpha, \beta) \equiv \exp[-z] \quad \text{(A.25)}$$

is just a normalization factor. Now that we are done with the general derivation, we add a subscript to the solution to distinguish it from the special cases of the IB and DIB:

$$q_\alpha(t|x) = \frac{1}{Z(x, \alpha, \beta)} \exp \left[\frac{1}{\alpha} \left(\log q(t) - \beta D_{\mathrm{KL}} \big[p(y|x) \mid q(y|t) \big] \right) \right]. \quad \text{(A.26)}$$

The IB solution is then

$$q_{\mathrm{IB}}(t|x) = q_{\alpha=1}(t|x) = \frac{q(t)}{Z} \exp \left[-\beta D_{\mathrm{KL}} \big[p(y|x) \mid q(y|t) \big] \right], \quad \text{(A.27)}$$

while the DIB solution is

$$q_{\mathrm{DIB}}(t|x) = \lim_{\alpha \to 0} q_\alpha(t|x) = \delta\big(t - t^*(x) \big), \quad \text{(A.28)}$$

with

$$t^*(x) = \operatorname*{argmax}_t \left(\log q(t) - \beta D_{\mathrm{KL}} \big[p(y|x) \mid q(y|t) \big] \right). \quad \text{(A.29)}$$

Acknowledgments

References

Atick, J. J., & Redlich, A. N. (1992). What does the retina know about natural scenes? *Neural Computation*, *4*(2), 196–210.

Bach, F. R., & Jordan, M. I. (2006). A probabilistic interpretation of canonical correlation analysis (Technical Report). Berkeley: University of California, Berkeley.

Barlow, H. B. (1981). Critical limiting factors in the design of the eye and visual cortex. *Proceedings of the Royal Society of London B: Biological Sciences*, *212*(1186), 1–34.

Barlow, H. B. (2001a). Redundancy reduction revisited. *Network Computation in Neural Systems*, *12*(3), 241–253.

Barlow, H. B. (2001b). The exploitation of regularities in the environment by the brain. *Behavioral and Brain Sciences*, *24*(4), 602–607.

Bekkerman, R., El-Yaniv, R., Tishby, N., & Winter, Y. (2001). On feature distributional clustering for text categorization. In *Proceedings of the 24th Annual ACM SIGR Conference on Research and Development in Information Retrieval* (pp. 146–153). New York: ACM.

Bekkerman, R., El-Yaniv, R., Tishby, N., & Winter, Y. (2003). Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research*, *3*, 1183–1208.

Blei, D. M., Griffiths, T. L., Jordan, M. I., & Tenenbaum, J. B. (2003). Hierarchical topic models and the nested Chinese restaurant process. In S. Thrun, L. K. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems, 16* (pp. 17–24). Cambridge, MA: MIT Press.

Chechik, G., Globerson, A., Tishby, N., & Weiss, Y. (2005). Information bottleneck for gaussian variables. *Journal of Machine Learning Research*, *6*(1532–4435), 165–188.

Cover, T. M., & Thomas, J. A. (2006). *Elements of information theory*. New York: Wiley-Interscience.

Creutzig, F., Globerson, A., & Tishby, N. (2009). Past-future information bottleneck in dynamical systems. *Physical Review E*, *79*(4), 041925–5.

Hecht, R. M., Noor, E., & Tishby, N. (2009). Speaker recognition by gaussian information bottleneck. In *Proceedings of the Annual Conference of the International Speech Communication Association*.

Hecht, R. M., & Tishby, N. (2005). Extraction of relevant speech features using the information bottleneck method. In *Proceedings of the Annual Conference of the International Speech Communication Association*.

Kandel, E. R., Schwartz, J. H., Jessell, T. M., Siegelbaum, S. A., & Hudspeth, A. J. (2013). *Principles of neural science* (5th ed.). New York: McGraw-Hill.

Kinney, J. B., & Atwal, G. S. (2014). Equitability, mutual information, and maximal information coefficient. *PNAS*, *111*, 3354–3359.

MacKay, D. J. C. (2002). *Information theory, inference and learning algorithms*. Cambridge: Cambridge University Press.

Olshausen, B. A., & Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, *381*(6583), 607–609.

Olshausen, B. A., & Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, *37*(23), 3311–3325.

Olshausen, B. A., & Field, D. J. (2004). Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, *14*(4), 481–487.

Palmer, S. E., Marre, O., Berry II, M. J., & Bialek, W. (2015). Predictive information in a sensory population. *Proceedings of the National Academy of Sciences*, *112*(22), 6908–6913.

Schneidman, E., Slonim, N., Tishby, N., de Ruyter van Steveninck, R. R., & Bialek, W. (2001). Analyzing neural codes using the information bottleneck method. In S. Becker, S. Thrun, & K.-R. Müller (Eds.), *Advances in neural information processing systems, 15*. Cambridge, MA: MIT Press.

Shamir, O., Sabato, S., & Tishby, N. (2010). Learning and generalization with the information bottleneck. *Theoretical Computer Science*, *411*(29–30), 2696–2711.

Simoncelli, E. P., & Olshausen, B. A. (2001). Natural image statistics and neural representation. *Annual Review of Neuroscience*, *24*(1), 1193–1216.

Slonim, N., & Tishby, N. (2000a). Agglomerative information bottleneck. In S. A. Solla, T. K. Leen, & K.-R. Müller (Eds.), *Advances in neural information processing systems* (pp. 617–623). Cambridge, MA: MIT Press.

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

Slonim, N., & Tishby, N. (2001). The power of word clusters for text classification. In *Proceedings of the European Colloquium on IR Research*.

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. (2004). How many clusters? An information-theoretic perspective. *Neural Computation*, *16*(12), 2483–2506.

Still, S., Crutchfield, J. P., & Ellison, C. J. (2010). Optimal causal inference: Estimating stored information and approximating causal architecture. *Chaos*, *20*.

Strouse, D., & Schwab, D. J. (2017). On the relationship between distributional and geometric clustering. Unpublished manuscript.

Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. In *Proceedings of The 37th Allerton Conference on Communication, Control, and Computing* (pp. 368–377).

Tishby, N., & Zaslavsky, N. (2015). *Deep learning and the information bottleneck principle*. arXiv.org.

Turner, R. E., & Sahani, M. (2007). A maximum-likelihood interpretation for slow feature analysis. *Neural Computation*, *19*(4), 1022–1038.

Wallace, G. K. (1991). The JPEG still picture compression standard. *Commun. ACM*, *34*(4), 30–44.