Survey of Rate Distortion and Information Bottleneck from the Perspective of Unsupervised Learning

Caleb Chuck

***Abstract*—This paper will introduce the ideas of unsupervised machine learning as motivation for presenting the underlying information theoretic theory from the perspective of channel coding, then transition into a discussion of various extensions of information bottleneck ideas and algorithms to applications. In particular, several examples of the optimization models applied to toy problems are used to demonstrate usage and equivalences between the various models. Though these examples only scratch the surface of the full range of ideas and implementations explored in this paper, they provide a motivation and background for such extensions. The paper concludes with an overview of the different applications of information theoretic concepts to learning techniques and data sets, relating various modern techniques to the informational bottleneck, and proposing several ideas for implementation.**

I. Motivation

UNSUPERVISED learning has found increasing usage due to the vast increase in unstructured information. Unsupervised learning involves the inference of some function which describes the structure of the data being analyzed. One general optimization for unsupervised learning is:

$$\min_f d(A, f(A))$$

$$\text{subject to}$$

$$\mathscr{C}(f(A)) \leq C$$

That is, minimize some distance metric $d(\cdot,\cdot)$ of $A_{n,m}$, the data matrix, and some function of A, subject to a constraint on the complexity $\mathscr{C}$ on the functional expression $f(A)$ being less than a value C. In this optimization problem, optimizing $f(A)$ is the unsupervised learning problem. In practice, three techniques have found broad usage: clustering and latent variable modeling (dimensionality reduction). For the example of low rank matrix approximation[1], the corresponding optimization is:

$$\min_{\tilde{A}} \|A - \tilde{A}\|_F^2$$

$$\text{subject to}$$

$$u_i \in \mathbb{R}^m, v_i \in \mathbb{R}^n$$

$$\tilde{A} = \sum_{i=1}^{n} \sigma_i u_i v_i^\top$$

That is, find a matrix that minimizes the Frobenius norm of the difference matrix between the data A and some other matrix $\tilde{A}$ being optimized, subject to the constraint that that rank of this other matrix must be less than some k. Notice that for A being dense, this will require fewer variables, for $k < \min(m,n)$, since this is described by: $km + kn + k$ terms, while in general the data matrix is described by mn terms. This problem is an example of dimensionalitiy reduction, since the dimension of the data has been reduced.

Clustering chooses some number of cluster centers, such that these cluster centers are representative of the data. That is, the generalized optimization formulation for clustering:

$$\min_{c_i \in \mathscr{C}} \sum_{i=1}^{n} d_1(a_i, \min_{c_i \in \mathscr{C}} d_2(a_i, c_i))$$

This is minimizing the distance metric d_1 between a_i, a data point in the data matrix, and the nearest cluster, where the nearest cluster is chosen by d_2. In general, $d_1 := d_2$. Clustering reduces the amount of data needed to describe a data matrix. This is done by replacing the data points with some representation of their closest cluster. Furthermore, clustering can provide information about the underlying structure of the data. Notice that this problem under fixed distance metric is in fact NP-complete for continuous $c_i \in \mathbb{R}^m$.

Clustering and dimensionality reduction parallels the information theoretic discussion of quantization in rate distortion and the accompanying informational bottleneck. Suppose that the data matrix A is produced from some source distribution P_A. Then in the limit of samples, P_A is the solution to the unsupervised learning problem, provided that it can be described with some finite complexity. This parallels the problem of sampling, where in order to describe a continuous, infinite signal, some number of samples are introduced, which can sometimes describe the signal[2]. In the rate distortion problem, the user attempts to describe some random process, which is possibly continuous, with a finite representation, subject to some distortion measure. Ultimately, this can be used to produce a clustering, using an estimated probability distribution from the data matrix, and mapping this possibly continuous distribution to some quantization distribution (of clusters). Then all points that are mapped to the same quantized value will be clustered.

[1] often implemented as PCA

[2] Particularly, under bandwidth constraints

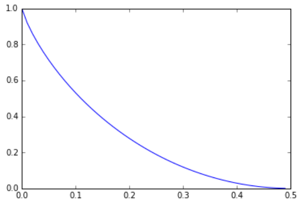


Figure II.1. Ideal Rate distortion values for Bernoulli .5 random variable

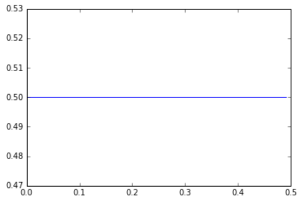


Figure II.2. Ideal prior probability values for Bernoulli .5 random variable

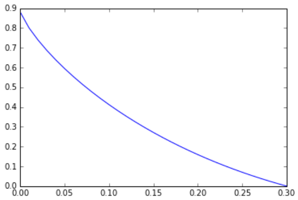


Figure II.3. Ideal Rate distortion values for Bernoulli .5 random variable

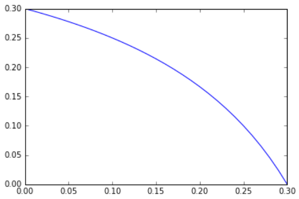


Figure II.4. Ideal prior probability values for Bernoulli .3 random variable

II. Background

Rate distortion theory is motivated from the description of a continuous random variable using some finite number of bits. It is clear that even a single real number requires an infinite number of bits to represent, and thus it will always be impossible to fully represent such a (random) variable. Thus, the optimization attempts to minimize the expected distortion for a given fixed rate. In a manner similar to channel coding, where a better rate is achievable by spreading out the bits over time, it turns out that "spreading out" the information over several random variables by their joint distribution leads to less distorted codes storing the same information, that is, quantizing a set of n iid random variables represented using nR bits, where R is the rate[1]. Some nomenclature involved with rate distortion theory:

- X, X_i: a random variable drawn from some distribution $\mathscr{X}$
- X^n: a sequence of random variables of length n
- $\tilde{X}$: the quantized representation of X, drawn from $\tilde{\mathscr{X}}$.
- $d(X,\tilde{X}) : \mathscr{X} \times \tilde{\mathscr{X}} \to \mathbb{R}^+$: The distortion function, which applies to the instances of $X, \tilde{X}$, $x, \tilde{x}$, and produces a distance. Some common distance metrices includes the hamming distortion: $\begin{cases} 0 & x = \tilde{x} \\ 1 & \text{otherwise} \end{cases}$, and norm-distance: $\|x - \hat{x}\|_2^2$. This extends to the sum of distances for a sequence
- $f_n : \mathscr{X}^n \to \{1, 2, \ldots, 2^{nR}\}$, the mapping of the input space into some quantized number of bits, $g_n : \{1, 2, \ldots, 2^{nR}\} \to \tilde{\mathscr{X}}^{\backslash}$, or the reconstruction alphabet. These function may be stochastic
- $\bar{d}(X,\tilde{X})$: the average distortion measure, which is evaluated as: $\sum\limits_{x^n} p(x^n) d(x^n, g_n(f_n(x^n)))$, $\sum\limits_{x,\tilde{x}} p(x)p(\tilde{x}|x)d(x,\tilde{x})$
- The rate distortion function $R(D)$ =the infimum of rate R such that the distortion D is achieved.

The rate distortion function solves the optimization problem:

$$\min_{p(\tilde{x}|x)} I(X;\tilde{X})$$

subject to

$$\sum_{x,\tilde{x}} p(x)p(\tilde{x}|x)d(x,\tilde{x}) = \bar{d}(X,\tilde{X}) \leq D$$

Note that the intuition of this is actually somewhat unusual, as this minimization takes the infimum over the probability distributions which destroy information about X. This is a result of the mathematics (see [1]), and this minimization results in the reduction of overall information needed to represent X.

This function can be directly applied to find the optimal probability distribution of some simple distributions. As an example, the distribution: $X \sim$ Bernoulli(p), and $\tilde{X} \in \{0,1\}$. Then:

$$I(X;\tilde{X}) = H(X) - H(X|\tilde{X}) \geq H(p) - H(Pr(X \neq \tilde{X})) \geq H(p) - H(D)$$

Notice the last term derives from the fact that the error term being sub-components of $\sum_{\tilde{x}=0,1} P(\tilde{x}) \sum_{x=0,1} P(x|\tilde{x}) \log(P(x|\tilde{x}))$, each of which is positive and thus increases the value. Notice that with a hamming distortion: $\sum_{x,\tilde{x}} p(x)p(\tilde{x}|x)d(x,\tilde{x}) = P(X \neq \tilde{X}) \leq D$.

Now, by demonstrating a code such that: $R(D) = H(p) - H(D)$, this demonstrates the optimality of this rate distortion function. The suggested code requires the property: $P(\tilde{X} = 1) = \tilde{p}, \tilde{p}(1-D) + (1-\tilde{p})D = p$, that is, the crossover probability reproduces: $P(X = 1) = p$. This resolves to: $\tilde{p} = \frac{p-D}{1-2D}$. Note that this solution only holds for $D \leq p$. Otherwise, the solution is trivial, where $\tilde{p} = 0$, which has distortion: $p \leq D$.

The rate distortion function, as well as optimal $\tilde{p}$ for bernoulli 0.5 random variable is illustrated in figure II.1 and figure II.2 respectively. Notice here that destroying the information equates minimizing the rate distortion, so the actually application of compression is hard to capture. In figure II.3, II.4, the rate distortion and optimal $\tilde{p}$ is shown for $X \sim$ Bernoulli .3. Notice that in this case, the rate distortion code produces $\tilde{p}$ with varying probabilities. As the distortion increases, $\tilde{p}$ probability decreases monotonically. This makes intuitive sense: as more damage is allowed, in the form of distortion or probability of error, the sparsity of the code increases. Notice also that though the alphabets of the codes are the same in this case, the sparsity of the code corresponds to greater compression. A code which less often has ones[3] can be represented with fewer bits over a bit sequence. In practice, some methods for sparsity compression include Huffman codes and reverse syndrome coding. These results demonstrate that destroying the information about the original signal does not simply confound the bits, as is the case with $p = .5$, but actually produces a more compressible signal in a monotonic way.

For more complex codes, it is more difficult to simply derive a solution to the optimization problem, though it is shown that for multivariate IID Gaussians, that is

$$X \sim \mathcal{N}(0, \begin{bmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_m^2 \end{bmatrix}$$

the distortion level for each the Gaussians follows a reversed water filling based on the magnitudes of their σ_i^2 [1]. For such cases, it is preferable to perform some algorithm on the probability distributions which begets the optimal quantization. While choosing the optimal alphabet to quantize over is a difficult problem, over a fixed alphabet, an intuitive algorithm exists for the solving of:

$$\mathscr{F}[p(\tilde{x}|x)] = I(X;\tilde{X}) + \beta\bar{d}(x,\tilde{x})$$

Or as an optimization problem:

$$\min_{p(\tilde{x}|x)} I(X;\tilde{X}) + \beta\bar{d}(X,\tilde{X})$$

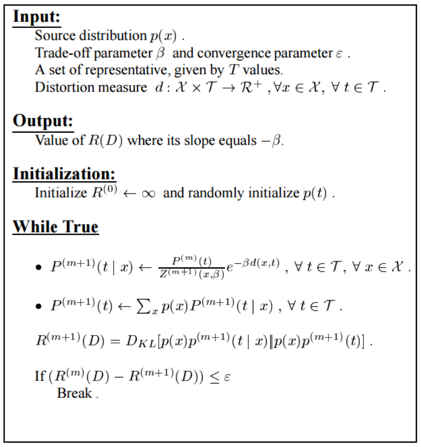
This formula is the relaxed lagrangian form of the original problem, where β is the chosen lagrange multiplier. Notice that solving this is not the same as solving the dual problem, which uses a parameter D for the distortion. Instead, the term $\beta\bar{d}(X,\tilde{X})$ regularizes the information confounding, since in order to minimize the value $I(X;\tilde{X})$, which compresses the signal, $\beta\bar{d}(X,\tilde{X})$ must also be minimized, thus producing a limit on the damage sustainable to the signal. Notice that this is an unconstrained minimization in $p(\tilde{x}|x)$. Thus, taking the derivative[2]:

$$0 = \frac{\delta\mathscr{F}}{\delta p(\tilde{x}|x)} =$$

$$p(x)\left[\log(p(\tilde{x}|x)/p(\tilde{x})) + 1 - \frac{1}{p(\tilde{x})}\sum_{x'} p(x')p(\tilde{x}|x') + \beta d(x,\tilde{x}) + \frac{\lambda(x)}{p(x)}\right]$$

Solving this for $p(\tilde{x}|x')$, noticing that: $\sum_{x'} p(x')p(\tilde{x}|x') = p(\tilde{x})$

$$\frac{p(\tilde{x}|x)}{p(\tilde{x})} = e^{\beta d(x,\tilde{x}) + \lambda(x)/p(x)}$$



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

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

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

While True

```
• $P^{(m+1)}(t \mid x) \leftarrow \frac{P^{(m)}(t)}{Z^{(m+1)}(x,\beta)} e^{-\beta d(x,t)} , \forall t \in \mathcal{T}, \forall x \in \mathcal{X}$ .
• $P^{(m+1)}(t) \leftarrow \sum_x p(x)P^{(m+1)}(t \mid x) , \forall t \in \mathcal{T}$ .
$R^{(m+1)}(D) = D_{KL}[p(x)p^{(m+1)}(t \mid x) \| p(x)p^{(m+1)}(t)]$ .
If $(R^{(m)}(D) - R^{(m+1)}(D)) \leq \varepsilon$
    Break .
```

Figure II.5. Concise description of the Blahut Arimoto algorithm, taken from [3]

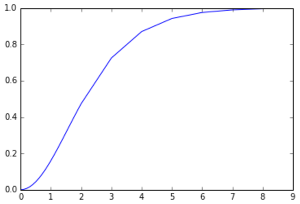


Figure II.6. Rate distortion values for Bernoulli 0.5 random variable, computed with the Blahut Arimoto algorithm

Noting that $e^{\lambda(x)/p(x)}$ is simply a normalization term gives: $p(\tilde{x}|x) = \frac{p_t(\tilde{x})}{Z_t(x,\beta)}\exp(-\beta d(x,\tilde{x}))$. The Z here are normalization terms.

From this, the Blahut Arimoto algorithm arises. Figure II.5 contains a concise description of the algorithm. The formulation of this algorithm is based on the dual minimization of the convex sets, $p(\tilde{x}), p(\tilde{x}|x)$, where $p(\tilde{x})$ arises from summing out the joint distribution, and notice that for the computation of the joint probability, the prior is needed. The proof of optimality of this algorithm is quite involved and found in [1], but it is important to note that because this algorithm is a dual minimization of convex sets, it will converge to a global optimal. This may not be the optimal rate distortion, recall, because the alphabet of $\tilde{x}$ may be limiting. Returning to the results from analysis of the Bernoulli $p = 0.5$ variable, Figures II.6, II.7 perform the same analysis, over varying β. Notice that β is anticorrelated with D, that is, smaller β corresponds to less penalty, which corresponds to higher D. In addition, the correlation between β and D with R is not the same, as changing β produces changing R with a different shape, which is expected because the Lagrangian relaxation is not likely to acct exactly as the original problem. Nonetheless, the results of β produce distributions which match those derived analytically. This fact agrees with the convexity of the optimized sets in practice. Similarly, figures II.8, II.9 provide analysis of the Bernoulli .3 distribution. Notice

[3] Thus more sparse

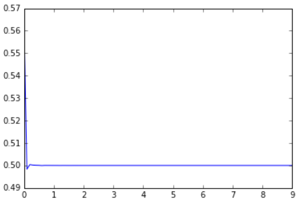


Figure II.7. Ideal prior probability values for Bernoulli 0.5 random variable, computed with the Blahut Arimoto algorithm

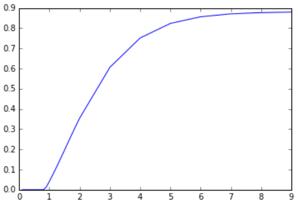


Figure II.8. Rate distortion values for Bernoulli 0.3 random variable, computed with the Blahut Arimoto algorithm

that for a given value of R, the prior probability of the distribution matches that of the ideal value, for the prior probabilities derived using the Blahut-Arimoto algorithm implementation.

III. Definition

THE information bottleneck attempts to alleviate the issue of choosing a distortion measure. Although in the Bernoulli .5 case studied here, the hamming distance can be chosen trivially with intuitive results, most probability distributions do not have this luxury. From the perspective of classification, the preferred choice of distortion is one which minimizes the damage to one‘s ability to classify the data points correctly[4]. In order to formalize this problem, denote X as some source distribution, Y as some class distribution, and $\tilde{X}$ as the quantization.

Note first that it is impossible to classify better than X. That is, if some perfect classification function $f(X) \to Y$ has a probability of error of ϵ, then it is impossible to do better than this, if all the information in X about Y is utilized. The impossibility of information gain about Y formalizes as: $I(\tilde{X};Y) \leq I(X;Y)$. Combining this with $I(\tilde{X};X)$ quantifying a degree of compression gives:

$$\min_{\tilde{X}} I(\tilde{X};X)$$

$$\text{Subject to}$$

$$I(\tilde{X};Y) \geq \mathscr{I}$$

This describes minimizing the information between $\tilde{X}$ and X. That is, maximizing the compression, while maintaining at least $\mathscr{I}$ bits of information about Y. Like with rate distortion codes, this relationship can also be expressed in a Lagrangian form:

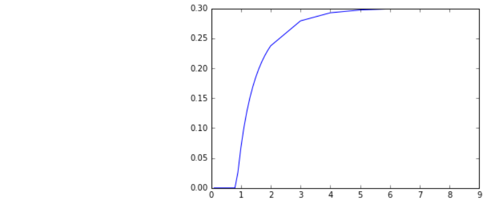
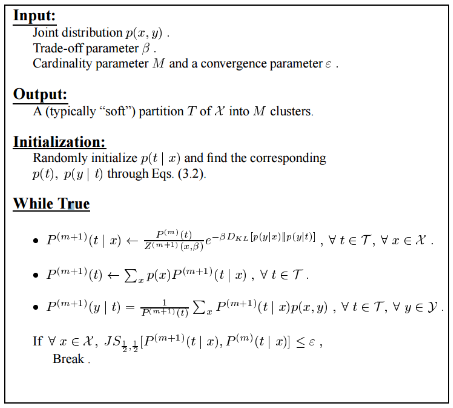


Figure II.9. Ideal prior probability values for Bernoulli 0.3 random variable, computed with the Blahut Arimoto algorithm



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

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

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

While True
    • $P^{(m+1)}(t \mid x) \leftarrow \frac{P^{(m)}(t)}{Z^{(m+1)}(x,\beta)} e^{-\beta D_{KL}[p(y|x) \| p(y|t)]}$ , $\forall\, t \in \mathcal{T}, \forall\, x \in \mathcal{X}$ .
    • $P^{(m+1)}(t) \leftarrow \sum_x p(x) P^{(m+1)}(t \mid x)$ , $\forall\, t \in \mathcal{T}$ .
    • $P^{(m+1)}(y \mid t) = \frac{1}{P^{(m+1)}(t)} \sum_x P^{(m+1)}(t \mid x) p(x,y)$ , $\forall\, t \in \mathcal{T}, \forall\, y \in \mathcal{Y}$ .
    If $\forall\, x \in \mathcal{X},\ JS_{\frac{1}{2},\frac{1}{2}}[P^{(m+1)}(t \mid x), P^{(m)}(t \mid x)] \leq \varepsilon$ ,
        Break .
```

Figure III.1. Concise description of the iterative information bottleneck method, as described in [3]

$$\min_{\tilde{X}} I(\tilde{X};X) - \beta I(\tilde{X};Y)$$

This form also allows for a series of self consistent equations, which are used to determine algorithms for solving the information bottleneck problem. Similarly, the mathematics involves in deriving the self-consistent equations is not particularly involved and can be reported here, first expressed in the paper [2].

Taking the derivative of the lagrangian gives, on the definition of Information gain: $\sum\limits_{x,\tilde{x}} p(\tilde{x}|x) p(x) \log(p(\tilde{x}|x)/p(\tilde{x}))$:

$$\begin{aligned}
\frac{\partial \mathscr{L}}{\partial p(\tilde{x}|x)} &= p(x)[1+\log(p(\tilde{x}|x))] \\
&\quad -\frac{\partial p(x)}{\partial p(\tilde{x}|x)}[1+\log(p(\tilde{x}))] \\
&\quad -\beta \sum_y \frac{\partial p(\tilde{x}|y)}{\partial p(\tilde{x}|x)} p(y)[1+\log(p(\tilde{x}|y))] \\
&\quad -\beta \frac{\partial p(\tilde{x})}{\partial p(\tilde{x}|x)}[1+\log(p(\tilde{x}))] - \lambda(x)
\end{aligned}$$

Which can be reduced to, by using the definition of

[4] Briefly, given inputs $\mathscr{X}$, and a reference variable $\mathscr{Y}$, the classification problem attempts to assign for each x_i, a data point in $\mathscr{X}$, some probability distribution on $P(y_i|x_i)$, where some training set X is given, which contains points x_i and their accompanying labels y_i

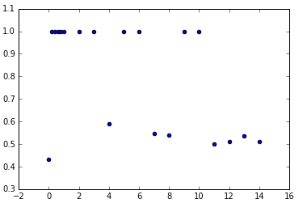


Figure III.2. A scatter plot of the prior probabilities given by the iterative information bottleneck

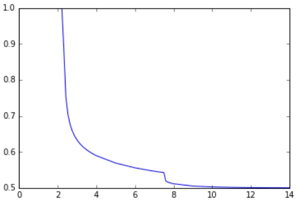


Figure III.3. Prior probabilities against increasing beta values, as computed by the information bottleneck algorithm, for prior probability 0.5

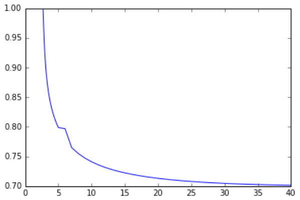


Figure III.4. Prior probabilities against increasing beta values, as computed by the information bottleneck algorithm, for prior probability 0.3

$p(\tilde{x}), p(\tilde{x}|y) = \sum_x p(\tilde{x}|x)p(x|y)$:

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial p(\tilde{x}|x)} &= p(x)[\log \frac{p(\tilde{x}|x)}{p(\tilde{x})} \\ &\quad +\beta \sum_y p(y|x)\log(\frac{p(y|x)}{p(y|\tilde{x})}) \\ &\quad -\frac{\lambda(x)}{p(x)} - \beta \sum_y p(y|x)\log\left(\frac{p(y|x)}{p(y)}\right)]\end{aligned}$$

Where the final term is a normalization factor. This then reduces to:

$$p(\tilde{x}|x) = \frac{p(\tilde{x})}{Z(x,\beta)}\exp\left(-\beta D_{KL}[p(y|x)|p(y|\tilde{x})]\right)$$

Here, D_{KL} is the kl divergence which arises as a replacement of the definition and not because of any particular assumptions. This equation suggests the equations suggested in figure III.1 are a concise statement of this algorithm. However, unlike rate distortion codes, the convergence of this algorithm on a global minima is not guaranteed, although the algorithm itself is guaranteed to converge somewhere. The intuition for the failures of the algorithm to converge lies in the three-way mutual convergence of $p(\tilde{x}|x), p(\tilde{x}), p(\tilde{y}|x)$, which are mutually dependent. The proof of this lies in [2]. In order to better understand these results, the algorithm in figure III.1 was implemented. Observe the results in figure III.2, which display results of the prior probability $\tilde{p}$ of the new distribution $\tilde{x}$, against increasing β. This simulation is set up with an input of a Bernoulli random variable with probability 0.5, with a dependency matrix:

$$\begin{bmatrix} & y=0 & y=1 \\ x=0 & 0.3 & 0.9 \\ x=1 & 0.7 & 0.1 \end{bmatrix}$$

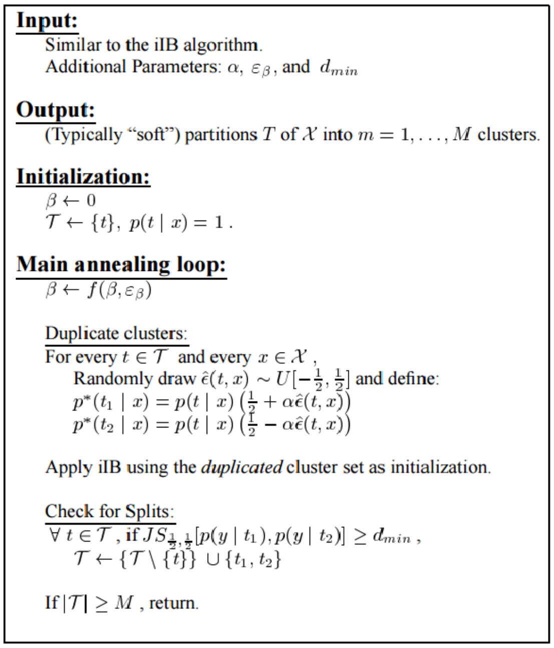
where the x values are on the rows. This dependency matrix has the property that x provides significant information about y, and thus by destroying the information about x, information about y is reduced. If the algorithm is run without significant information between x and y, the results will simply be a complete compression of x, where $\tilde{p} = 0$. Notice that because the way the information is preserved is arbitrary, it is also possible for: $\tilde{p} = 1$. In either case, the distribution has reduced into a single point.

Returning to figure III.2, note that the results of single iterations run with the IB algorithm often converges to the local optimum of compression the value to a single point. In fact, one weakness of this algorithm is that certain values of β will be more likely to converge to local optima, which in this example is at $\tilde{p} = 1$. Nonetheless, for iterations with β, it is clear that the prior probability is converging to preserve more information with increasing beta. In this case, this correlates with increased dependence on x. However, to better understand the trade off between compression and preservation, it is necessary to observe more samples. In figure III.3, the IB algorithm was run 30 times for each value of β, and the minimum of the posterior probabilities are plotted. Note the smooth transition from having no information preserved at $\beta = 2$ to having most of the information preserved at $\beta = 6$. Additional values of β were computed to understand the smoothness of the graph. Note that this curve strongly resembles the asymptotic of $\frac{1}{\beta}$, and additionally, does not follow exactly the same curve as Rate distortion codes. Nonetheless, there is a monotonic tradeoff between the preservation of information about Y, and the compression of X, for increasing β. Figure III.4 provides more insight, using another tradeoff curve, if the prior probability of X, p, is 0.3 instead of 0.5. Note that the tradeoff of information was significantly lower, which is most likely the result of there being less overall information between Y and X, since $X = 1$ preserves more information than $X = 0$. Finally, note these simulations demonstrate relationships between X, Y for what is essentially a binary asymmetric channel[5].

While the algorithm suggested in figure III.1 is the simplest conclusion of the self-consistent equations, the structure of the solutions suggests that:

$$\frac{\partial I(\tilde{X},Y)}{\partial I(X,\tilde{X})} = \beta^{-1}$$

[5] Note that the information stored about Y was not shown here, but can be observed by running the code at https://github.com/CalCharles/InfoBottle121.git



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

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

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

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

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

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

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

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

Figure III.5. A concise description of the annealing algorithm for computing the relaxed information bottleneck problem, as described in [3]

This states that the amount of information preserved about Y, compared to the amount of information removed about X, is related by the regularization term. Notice that in the figure, this property is not described, although by running the simulations it can be verified. The figures rely on this to demonstrate how the destruction of information for a given β also describes the information preserved about Y, since the rate of change is a constant. Perhaps a more interesting result of this is that it suggests a simulated annealing algorithm, shown in Figure III.5, which alters the value of β based on detecting values in $\tilde{X}$ for which the value "splits," or takes on multiple values of Y with similar probability, then moving to increasing β as increasing "splits" occur. This then gives a structure for which varying values of β result in varying amount of split. The demonstration of how the algorithm achieves this is located in [3]

IV. Application

MANY applications of the information bottleneck theorem arose, especially due to the existence of relatively implementable algorithms such as those shown in figures III.1, III.5. Furthermore, for multivariate Gaussian variables (As with the water filling result from rate distortion codes) the solution to the information bottleneck problem is closed form. Rather than a discrete clustering, or quantization, solution, this solution results in a continuous representation [4]. The intuition for this continuous solution results from the definition of multivariate Gaussian variables using means means and co-variances. The result derives simply from the continuous derivative of Gaussian entropy, when applied to Gaussian random variables $X, \tilde{X}, Y$. The derivation is in[4], with the result as:

$$\tilde{X} = \begin{cases} \begin{bmatrix} \mathbf{0}^\top & \ldots & \mathbf{0}^\top \end{bmatrix} & 0 \leq \beta \leq \beta_1 \\ \begin{bmatrix} \alpha_1 \mathbf{v}_1^\top & \mathbf{0}^\top & \ldots & \mathbf{0}^\top \end{bmatrix} & \beta_1 \leq \beta \leq \beta_2 \\ \begin{bmatrix} \alpha_1 \mathbf{v}_1^\top & \alpha_2 \mathbf{v}_2^\top & \mathbf{0} & \ldots & \mathbf{0}^\top \end{bmatrix} & \beta_2 \leq \beta \leq \beta_3 \\ \vdots \end{cases}$$

Where $\mathbf{v}_1^\top, \ldots, \mathbf{v}_n^\top$ are the left eigenvectors of:

$$\Sigma_{x|y} \Sigma_x^{-1}$$

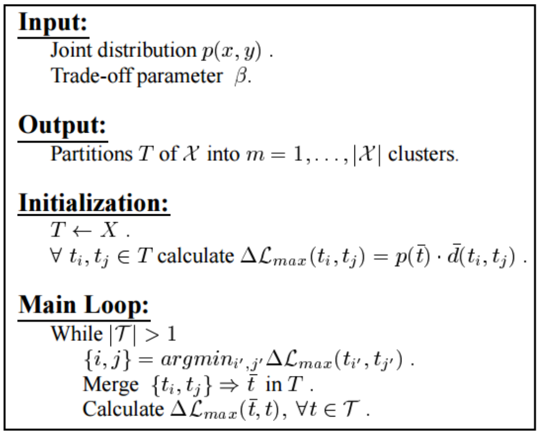
and

$$\beta_i = \frac{1}{1 - \lambda_i}, \alpha_i = \sqrt{\frac{\beta(1 - \lambda_i) - 1}{\lambda_i r_i}}, r_i = \mathbf{v}_i^\top \Sigma_x \mathbf{v}_i$$

This result demonstrates as expected, that β provides a measure of the amount of information stored about the data in $\tilde{X}$, where increasing β suggests an increasing amount of information. Then, for this problem, the solution appears to take on properties similar to one for principle component analysis, where the number of principle components is increased with increasing necessity for data[6]. Furthermore. this solution also suggests a way to relate the labels of the training set with the data to perform an improved version of the dimensionality reduction, or choose an optimal number of decomposition components. Finally, this method provides a similarity between the analysis of continuous representations of $X, Y, \tilde{X}$ with discrete ones: $X, Y, \tilde{X}$.

Another interesting application of the information bottleneck approach involves the application of the so called double clustering and agglomerative information bottleneck. In practice, one of the significant difficulties of using the information bottleneck involves the problem of determining the probability distribution of $P(X, Y)$, or the related problem: $P(X)$. For example, the problem of choosing a feature space for "small" images, such as those in the MINST dataset, involves determining the distribution of values for $784 \cdot n$ bits, where n is the number of bits used to represent a real number. Even if only one bit is used to represent each pixel, this would still involve an infeasible amount of memory, when considering all of the combinations: 784^{2^n}, simply to determine $P(X)$, the prior distribution. It is viable to apply Gaussian assumptions to the distribution, but this is often too strong. In digit data, such an assumption states that all symbols follow a derived $l2$ distance metric from each other, which is not often true. For other learning problems, Gaussian assumptions often fail to perform well. Similarly, using a naive Bayes model can reduce the complexity to manageable levels, but this makes strong independence assumptions which are generally not true. For the example of images, this would make the pixels independent, but this is clearly false, as images contain a great deal of symmetry, which are often the structures searched for in computer

[6] Notice that this observation is made based on the understanding of principle components as singular values, which are directly related to the eigenvalues of the empirical covariance matrix, or the inverse covariance matrix incorporated with inverse eigenvalues



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

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

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

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

Figure IV.1. A concise description of the agglomerative algorithm for computing the "hard" information bottleneck problem, taken from [3]

vision. Thus, this problem is addressed in the agglomerative algorithm. This algorithm is concisely described in Figure IV.1. In essence, it describes putting a cluster point at each of the given data inputs, then subsequently combining the data points based on the lowest information loss. Notice that this still requires an approximation of the probability distribution, but, as outlined in [3], the criterion on merging need only be performed at a local level, by the following metric:

$$JS_{\Pi}[p(y|\tilde{x}_i), p(y|\tilde{x}_j)] - \beta^{-1} JS_{\Pi}[p(x|\tilde{x}_i), p(x|\tilde{x}_j)]$$

However, note that while this is substantially smaller than containing a full matrix of all $|\tilde{X}|$ and $|X|$ comparisons, this is still substantially too large for true continuous variables. Nonetheless, [5] demonstrates this technique being used with the clustering of documents based on their word occurrences, performing an agglomerative technique on the different kinds of words. Note this does require an independence assumption on the words to be feasible.

The double clustering technique shown in [5] first clusters on the words, a much larger and harder to cluster set based on the agglomerative method, which results in a quantization of the word space into a much smaller word cluster space. Then, the document space is clustered based on the word clusters contained by the documents. The number of documents is much smaller, allowing for full conditional distributions based on the word clusters. This double clustering method works well compared with other clustering techniques for the work of classifying documents, as demonstrated in [5], when compared with other classifying techniques. This largely makes sense, since the quantization defined by IB in general takes into account better the information about the reference variable Y. The idea of clustering the data points by some metric to a quantization small enough to derive an empirical conditional distribution is discussed in the subsequent Discussion section.

This section concludes with the application of the information bottleneck to a Markovian clustering model, as described in [6]. This understanding follows similarly to the agglomerative clustering algorithm in [5]. Here, the data points are are assigned to nodes on a Markov chain graph, and the distances[7] between the data points assigned based on

$$\exp(-\lambda d(\mathbf{x}_i, \mathbf{x}_j))$$

This distance metric is a sort of exponential pairwise transition probability. This matrix will have some initial distribution based on the values of the data points, and to find the distribution after a t step random walk, the matrix will have the form:

$$P^t x_0$$

where P is the transition matrix defined above. Then, based simply on the properties of the Markov chain, notice that the information about some variable y will be stored in the initial distribution of this Markov chain. As the Markov chain is allowed to decay toward the stationary distribution, more information about the initial distribution will be lost. If the distance metric of:

$$d(\mathbf{x}_i, \mathbf{x}_j) = (x_i - x_j)^2 + (y_i - y_j)^2$$

is chosen, then the rate of information lost will be based on the bottleneck formed by preserving information about y. In [6], it is shown that structures appear as the stationary distribution is approached, which resemble the true clusters of the graph. This algorithm provides a link between clustering and semi-stable structures in a Markov chain, and does not need to quantify the underlying probability space of the data points, which makes it easier to implement.

V. Discussion

Returning from the implementation of particular clustering schemes based directly on the information bottleneck to the general problem of machine learning, this section will discuss the application of information theoretic understandings applied to existing algorithms. Rate distortion provides a general framework for understanding quantities about the structure of the data. In clustering, this often involves the amount of information preserved about the overall data. The information bottleneck approach augments this to be the amount of information preserved about the reference variables. In [7], artifacts in the rate distortion suggest an ideal number of clusters. This is intuitive, since if some number of clusters exists, it should be that if those clusters are found, any excess of description would not lead to significant change in distortion. This section explores such applications as [7], of both the information bottleneck and rate distortion codes.

One such extension of the information bottleneck is explored in [8]. Unsupervised learning attempts to derive structures in the data based on some criterion. However, not all of the side structures are equally important, and dynamics of the system, or soft knowledge might demonstrate that it is preferable for some structures to be ignored. One

[7] Recall for a Markov chain that distances involves the probability of transitioning between nodes of the Markov chain

such example of this might be missing data in click-through prediction. Often, sub-fields of the ad “impression” data is missing[8]. However, a classifying model might classify based on this information even though it may not be preferable to sustain the interpretability of the classifier.

The problem of side information relates also to rate distortion, where if some mutual variable W which contains information about X is provided at both ends, it is preferable to avoid transmitting information about W resulting in:

$$\min_{\bar{d}(X,\tilde{X})\leq D} \quad I(X;\tilde{X})-I(\tilde{X};W)$$

as described in [9]. The problem of side information involves a simple extension of the Lagrangian of the information bottleneck:

$$\mathscr{L}=I(X;\tilde{X})-\beta[I(\tilde{X};Y^{+})-\gamma I(\tilde{X};Y^{-})]$$

Y^{-} is the irrelevance variable in this case. Notice that this augmentation attempts to remove information about the irrelevance variable, by giving it the same sign as $I(X;\tilde{X})$, the compressor. In [8], this problem is formulated into a new set of self-consistent equations, producing a new iterative algorithm which can be used for text classification, similar to the problem described in [5]. The irrelevance information in this paper involved structures of the data common to all documents, and thus not useful for differentiation.

The information bottleneck theory also appears commonly in the analysis of networks and graphs. In the case of structured graphs, such as the neural network, the information bottleneck framework provides a theoretical basis for understanding the transition of un-interpretable structures in the data to highly compressed labels. Neural nets have proven particularly useful at performing tasks humans can do well: interpreting images and sound data. In these problems, almost all of the information about the relevance variable is in some way contained in the original input signal X. A neural net consists of several alternating layers of linear maps and non-linearity functions. At each non-linearity, it is possible to determine the amount of information about X remaining, given a distribution. Intuitively, each layer must be bounded in information, that is, for layer X_i

$$I(X;X_i)\leq I(X;X_{i-1}, I(Y;X_i)\leq I(Y;X_{i-1})$$

However, in the case of a highly accurate neural net, it appears that this function maintains the information between $I(X_i;Y)$, while reducing $I(\tilde{X};X_i)$, at least at the output layer. In fact, neural nets have been used as compressors, by forcing the data through a small “bottleneck” of nodes [10]. For the case of neural nets [11] provides a theoretical framework for exploring the information stored in the layers relative to the optimal information to rate ratio.

Finally, The information bottleneck method also brings up the interesting idea when applied to a data processing problem. Imagine the problem shown in figure V.1, where a single operator performs some learning algorithm, but lacks the memory to contain of the data. Since the information bottleneck provides a theoretical framework for estimating the relevance of a compression of X toward the computation of Y, it may provide a framework for determining some stochastic combination of the input data such that when the operator receives the data, the training set provides maximal information about the relevant structures in Y. In fact, modern methods currently suggest that random projections of the data[9], can be effective for the convergence of convex optimization algorithms, which are commonly the workhorse for machine learning[10] [12].

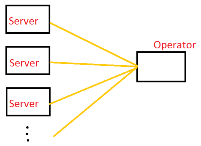


Figure V.1. Each of the servers sends some of its data to the operator, such that the operator then performs some machine learning algorithm. notice the primary weakness here is that the operator does not have the same kind of storage as the servers, and thus can only receive a small subset of the total data.

VI. Conclusion

THE subject of the information bottleneck provides a rich set of methods for studying learning problems in general, even going beyond the problem of unsupervised learning which motivated it. In recent years, less work has been dedicated to the study of new algorithms directly as a result of the information bottleneck, instead understanding it as a bridge between the statistical problem of machine learning and the information theoretic problem of rates. The weaknesses of the information bottleneck, and rate related problems in general in this domain, involves either the assumption of probability priors on continuous variables, or the difficulty of deriving a useful empirical distribution. However, in conjunction with modern algorithms, the information bottleneck provides a framework by which to study and quantify the usefulness of a particular algorithm or heuristic to new sets of data, as well as deriving new algorithms and heuristics based on the theoretical optimization of the information bottleneck.

More practically, the information bottleneck provides a quantified understanding of the primary problem in dealing with the vast quantity of data now present. Videos, audio, images, sensor data and text, which comprise a vast proportion of the data accessible to be understood by machine learning, all contain compressed representations when related to their use. For example, a security video might contain hours of footage, but only a small amount is useful for law enforcement. A long document contains large portions of example text, but the meat of the information is a single intuition. Ultimately, the information bottleneck provides a concise framework for the engineering problem

[8] Suppose that the user has used more private browser settings

[9] Take the matrix product of the data with a randomly generated matrix

[10] note that the support vector machine is a convex program

of producing devices which can distinguish the signal from this chatter.

References

[1] T. M. Cover and J. A. Thomas, *Elements of Information Theory* (Wiley, New York, 1991)

[2] Tishby, Naftali, Fernando C. Pereira, and William Bialek. “The information bottleneck method.” arXiv preprint physics/0004057 (2000).

[3] Slonim, Noam. “The information bottleneck: Theory and applications.” Diss. Hebrew University of Jerusalem, 2002.

[4] Chechik, Gal, et al. "Information bottleneck for Gaussian variables." Journal of Machine Learning Research. 2005.

[5] Slonim, Noam, and Naftali Tishby. "Document clustering using word clusters via the information bottleneck method." Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2000.

[6] Tishby, Naftali, and Noam Slonim. "Data clustering by markovian relaxation and the information bottleneck method." NIPS. 2000.

[7] Sugar, Catherine A., and Gareth M. James. "Finding the number of clusters in a dataset." Journal of the American Statistical Association (2011).

[8] Chechik, Gal, and Naftali Tishby. "Extracting relevant structures with side information." Advances in Neural Information Processing Systems. 2002.

[9] Wyner, Aaron D., and Jacob Ziv. "The rate-distortion function for source coding with side information at the decoder." Information Theory, IEEE Transactions on 22.1 (1976): 1-10.

[10] Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." Science 313.5786 (2006): 504-507.

[11] Tishby, Naftali, and Noga Zaslavsky. "Deep learning and the information bottleneck principle." Information Theory Workshop (ITW), 2015 IEEE. IEEE, 2015.

[12] Pilanci, Mert, and Martin J. Wainwright. "Randomized sketches of convex programs with sharp guarantees." Information Theory, IEEE Transactions on 61.9 (2015): 5096-5115.