A Rate-Distortion One-Class Model and its Applications to Clustering

Koby Crammer CRAMMER@CIS.UPENN.EDU
Partha Pratim Talukdar PARTHA@CIS.UPENN.EDU
Department of Computer & Information Science, University of Pennsylvania, Philadelphia, PA 19104, USA

Fernando Pereira[1] PEREIRA@GOOGLE.COM
Google, Inc. 1600 Amphitheatre Parkway, Mountain View, CA 94043, USA

Abstract

In one-class classification we seek a rule to find a coherent subset of instances similar to a few positive examples in a large pool of instances. The problem can be formulated and analyzed naturally in a rate-distortion framework, leading to an efficient algorithm that compares well with two previous one-class methods. The model can be also be extended to remove background clutter in clustering to improve cluster purity.

1. Introduction

Often we are given a large set of data items among which we would like to find a coherent subset. For instance, in document retrieval we might want to retrieve a small set of relevant documents similar to a few seed documents. In genomics, it is useful to find the set of genes that are strongly co-expressed with a few genes of interest. In both cases, we prefer high-precision answers over high-recall ones.

A popular intuition for this *one-class classification* problem is that of finding a small ball (under some appropriate norm) that contains as many of the seed elements as possible (Tax & Duin, 1999). Most previous approaches to the problem take the point of view of outlier and novelty detection, in which most of the examples are identified as relevant. However, Crammer and Chechik (2004) seek a small subset of relevant examples, rather than keep all but few outliers.

Most approaches to one-class classification use convex cost functions that focus on the large-scale distribution of the data. Those functions grow linearly outside class and and are constant inside it (Schölkopf et al., 1995; Tax & Duin, 1999; Ben-Hur et al., 2001). In a related study, Schölkopf et al. (2001) seek to separate most of the examples from the origin using a single hyperplane. More recently, Crammer and Singer (2003) generalized that approach to the general case of Bregman divergences. In all of those methods, the convexity of the cost function forces the the solution to shrink to the center of mass as the radius of the ball goes to zero, thus ignoring any local substructure.

In contrast to the previous work, Crammer and Chechik (2004) assumed that the distribution of points outside the one class is not relevant, so they chose a cost function that grows linearly inside the class but is constant outside it. This cost function is thus indifferent to the values of the irrelevant instances. A flat cost outside the class is expected to be better than a growing cost when the relevant instances are mostly in a small region, or when there are relatively few relevant instances. Unfortunately, their cost function leads to a non-convex optimization problem that requires an approximate solution.

Using ideas from rate-distortion theory (Cover & Thomas, 1991), we express the one-class problem as a lossy coding of each instance into a few possible instance-dependent codewords. Unlike previous methods that use just two (Crammer & Chechik, 2004) or a small number (Bekkerman & McCallum, 2005) of possible codewords for all instances, the total number of codewords in our method is greater than the number of instances. To preclude trivial codings, we force each instance to associate only with a few possible codewords. Finding the best coding function is an optimization problem for which we provide an efficient algorithm. The optimization has an "inverse temperature" parameter that represents the tradeoff between compression and distortion. As temperature

[1] Work done mainly at the University of Pennsylvania.

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

decreases, the solution passes through a series of phase transitions associated with different sizes for the one class. This model outperforms two previous algorithms proposed for the problem, which are effective only in more restricted situations.

Our one-class model is also effective on the task of clustering a set of instances into multiple classes when some of the instances are clutter that should not be included in any cluster. This task can be reduced to an alternation between applications of the one-class algorithm and hard clustering. Initial experiments with synthetic and real world data show that by leaving some instances out of the clusters, the quality of the clustering improves.

2. One-Class as Rate-Distortion Optimization

We are given a set of instances indexed by the integer random variable $1 \leq X \leq m$. Each instance is described by a *point* $\mathbf{v}_x \in \mathbb{R}^d$ (possibly restricted to the simplex), and $p(x) = p(X = x)$ is a prior distribution over instances. Our goal is to find a *small coherent* subset of instances from a large set of possible instances. In particular, the learning task is to find a *centroid* $\mathbf{w}$ in the space such that there are many seed instances $\mathbf{v}_x$ close to it.

We formalize the task as a source coding problem. An instance x is either coded with the one class, with distortion $\mathcal{D}\left(\mathbf{v}_x \| \mathbf{w}\right)$, and assigned the code 0, or it is coded as itself with zero distortion. The distortion $\mathcal{D}$ can be any Bregman divergence (Censor & Zenios, 1997), which includes as special cases the Euclidean distance and the KL divergence between distributions.

The random variable T represents the code for an instance: if $T = 0$, the instance was coded with the one class, while if $T = x > 0$, the instance is coded as itself. Although T has $m + 1$ distinct values, only one code x is associated with the event $T = x > 0$. The coding process is summarized by the conditional probability $q(t|x)$ of encoding x as t. These constraints mean that $q(t|x) = 0$ if $t \notin \{x, 0\}$, that is, the only nonzero probability outcomes for x are $T = 0$ or $T = x$.

The marginal

$$q(0) = \sum_x p(x)q(0|x) \ , \tag{1}$$

is the probability of assigning any instance to the one class. The other marginals are a product of two terms $q(x) = p(x)q(x|x)$, because of the constraints that $q(t|x) = 0$ for $t \neq 0$ and $t \neq x$. We explicitly allow soft assignments, $0 \leq q(0|x) \leq 1$. As we will see below, instances in the one class have a hard assignment to the class, but instances outside have a soft assignment.

We use the information bottleneck (IB) framework (Tishby et al., 1999) to formalize the assignment process. IB is an information-theoretic approach to regularized unsupervised learning that aims to extract a meaningful representation of some data X based on its association with side information. For generality, we choose here the rate-distortion formulation of the IB, which solves for the assignment by optimizing the tradeoff between two quantities: the amount of compression applied to the source data X, measured by the mutual information $I(T;X)$, and the average distortion between the data and its representation:

$$\min_{\mathbf{w},\{q(0|x)\}} I(T;X) + \beta D(\mathbf{w}, \{q(0|x)\}) \ . \tag{2}$$

For one-class learning, the distortion term measures how well on average the centroid $\mathbf{w}$ serves as a proxy to each of the instances $\mathbf{v}_x$:

$$D(\mathbf{w}, \{q(0|x)\}) = \sum_x p(x)q(0|x)\mathcal{D}\left(\mathbf{v}_x \| \mathbf{w}\right) \ .$$

In contrast with standard rate distortion and IB formulations, the average distortion is computed only for $T = 0$, because the distortion is zero for $T > 0$.

We first rewrite the mutual information term using the constraints $q(t|x) = 0$ if $t \neq x$ and $t \neq 0$:

$$\begin{aligned} I(T;X) &= \textstyle\sum_{x,t} p(x)q(t|x)\log\left(\frac{q(t|x)}{q(t)}\right) \\ &= \textstyle\sum_x p(x)\left[q(0|x)\log\left(\frac{q(0|x)}{q(0)}\right) + q(x|x)\log\left(\frac{q(x|x)}{q(x)}\right)\right] \\ &= \textstyle\sum_x p(x)\left[\begin{array}{l} q(0|x)\log\left(\frac{q(0|x)}{q(0)}\right) + \\ (1-q(0|x))\log\left(\frac{q(x|x)}{q(x|x)p(x)}\right) \end{array}\right] \ . \end{aligned}$$

Then, the minimization (2) can be written as:

$$\begin{aligned} \min_{\{q(t|x),\mathbf{w}\}} \quad & \sum_{x=1}^{m} p(x)\left[q(0|x)\log\left(\frac{q(0|x)}{q(0)}\right)\right. \\ & \qquad \left. + (1-q(0|x))\log\left(\frac{1}{p(x)}\right)\right] \\ & \qquad + \beta\sum_x p(x)q(0|x)\mathcal{D}\left(\mathbf{v}_x \| \mathbf{w}\right) & (3) \\ \text{s.t.} \quad & 0 \leq q(0|x) \leq 1, 1 \leq x \leq m & (4) \end{aligned}$$

The corresponding Lagrangian is:

$$\textstyle\sum_{x=1}^{m} p(x)\left[q(0|x)\log\left(\frac{q(0|x)}{q(0)}\right) + (1-q(0|x))\log\left(\frac{1}{p(x)}\right)\right] \\ + \beta\textstyle\sum_x p(x)q(0|x)\mathcal{D}\left(\mathbf{v}_x \| \mathbf{w}\right) + \textstyle\sum_x p(x)\nu_x q(0|x) \ .$$

Setting to zero its derivative with respect to $q(0|x)$, we get:

$$p(x)\left[\log\left(\frac{q(0|x)}{q(0)}\right)+\beta p(x)\mathcal{D}\left(\mathbf{v}_x\|\mathbf{w}\right)+\log p(x)+\nu_x\right]=0 \ ,$$

Using the KKT conditions, we solve for $q(0|x)$:

$$q(0|x) = \min\left\{q(0)\frac{e^{-\beta\mathcal{D}(\mathbf{v}_x\|\mathbf{w})}}{p(x)}, 1\right\} \ . \qquad (5)$$

Setting the derivative of the Lagrangian with respect to $\mathbf{w}$ to zero we get:

$$\mathbf{w} = \frac{\sum_x p(x)q(0|x)\mathbf{v}_x}{\sum_x p(x)q(0|x)} = \sum_x q(x|0)\mathbf{v}_x \ . \qquad (6)$$

That is, the centroid is the average of all the points $\mathbf{v}_x$ weighted by their probability of membership in the single class.Like in the IB, the solution has a set of self-consistent equations: (1), (5), and (6).

Note that this rate-distortion formulation can be expressed as a tradeoff between two information quantities, as in the original IB. When $\mathcal{D}$ is the KL divergence, the optimization in (2) (or (3)) is equivalent to minimizing the tradeoff $I(X;T) - \beta I(T;Y)$ under the above constraints, where the random variable Y gives side information through the vectors $\mathbf{v}_x$.

3. Algorithm

The sequential algorithm of Slonim (2003) finds efficiently a local maximum of the IB objective. The algorithm alternates between selecting an instance and deciding whether moving it to another cluster would improve the objective. The item is reassigned to the cluster which yields the best improvement. A similar algorithm has been proposed for one-class problems (Crammer & Chechik, 2004). At each round, an instance is either removed from the class or added to the class, depending on what would most improve the objective.

We present a different algorithm for our model, inspired by Blahut-Arimoto algorithm and the original IB algorithm. The new algorithm iterates between the self-consistent equations (1), (5), and (6). Analogously to those algorithms, ours alternates between fixing $q(0|x)$ and $q(0)$ and fixing $\mathbf{w}$, and solving for the other parameters. We solve easily for $\mathbf{w}$ by computing the weighted average in (6). To solve for $q(0|x)$ and $q(0)$, let $\mathbf{w}$ be fixed and define $d_x = \mathcal{D}\left(\mathbf{v}_x\|\mathbf{w}\right)$. We now show how to compute $q(0|x)$ and $q(0)$ efficiently.

Eq. (5) cannot be solved directly for $q(0|x)$ because it involves $q(0)$, which in turn depends on $q(0|x)$. However, we can break this cycle by analyzing more carefully the properties of the solution. Let $\mathcal{C} = \{x : q(0|x) = 1\}$. From (1) we get:

$$q(0) = \sum_x p(x)q(0|x) = \sum_{x\in\mathcal{C}} p(x) + q(0)\sum_{x\notin\mathcal{C}} e^{-\beta d_x} \quad (7)$$

Assume that $\mathcal{C} \neq \emptyset$. Solving for $q(0)$, we obtain: $q(0) = \left(\sum_{x\in\mathcal{C}} p(x)\right) / \left(1 - \sum_{x\notin\mathcal{C}} e^{-\beta d_x}\right)$. This equation is well defined if $0 \leq q(0) \leq 1$, or equivalently:

$$\sum_{x\notin\mathcal{C}} e^{-\beta d_x} \leq 1 - \sum_{x\in\mathcal{C}} p(x) \ . \qquad (8)$$

If $\mathcal{C}$ contains all the points, this is trivially satisfied. If $\mathcal{C} = \emptyset$, (7) becomes $q(0)\left(1 - \sum_x e^{-\beta d_x}\right) = 0$. Therefore, there is a unique β_0 such that for all $\beta \geq \beta_0$ we have $q(0) = 0$. If $p(x) > 0$ for all x we then have $q(0|x) = 0$.

In summary, the solution of the optimization problem is given by the set $\mathcal{C} = \{x : q(0|x) = 1\}$. We cannot search for that set naïvely, but fortunately the following lemma gives an efficient way to find the set by sorting its possible members.

Lemma 1 *Let* $x_1, \ldots, x_m$ *be a permutation of* $[1, m]$ *such that* $0 < \beta d_{x_1} + \log p(x_1) \leq \cdots \leq \beta d_{x_m} + \log p(x_m)$. *Then* $\mathcal{C} = \{x_i : 1 \leq i \leq k\}$ *for some* $k \in [0, m]$.

Proof: Assume that $\mathcal{C} \neq \emptyset$. From (5) we know that $q(0|x) = \min\left\{q(0)e^{-\beta d_x - \log(p(x))}, 1\right\}$. We now show that if $x_k \in \mathcal{C}$ for some k, then $x_j \in \mathcal{C}$ for all $1 \leq j < k$. If $x_k \in \mathcal{C}$, by definition $q(0)e^{-\beta d_{x_k} - \log p(x_k)} \geq 1$. For $j < k$, by hypothesis we have $-\beta d_{x_k} - \log p(x_k) \geq -\beta d_{x_j} - \log p(x_j)$. Thus, $q(0)e^{-\beta d_{x_j} - \log p(x_j)} \geq 1$, and thus $x_j \in \mathcal{C}$. ■

The lemma allows us to solve (3) easily for a fixed $\mathbf{w}$. The inputs for the algorithm are the prior over items $p(x)$, the distortions d_x, and the tradeoff parameter β. First, we order the items x in ascending order of the combined distortion and log-prior $\beta d_x + \log(p(x))$. As in the lemma, we obtain an ordering $x_1, \ldots, x_m$. Among the possible $\mathcal{C} = \{x_i : 1 \leq i \leq k\}$ that satisfy (8), we choose the one that minimizes the objective. A naïve implementation would require $\mathcal{O}(m \log m)$ time to sort the items, and then additional $\mathcal{O}(m)$ steps for each of the m candidate subsets $\mathcal{C}$, yielding an overall complexity of $\mathcal{O}(m^2)$. However, we can use dynamic programming to compute the objective for $\mathcal{C} \cup \{x_k\}$ from quantities saved from computing the objective for $\mathcal{C}$. Equation (3) can be rearranged as:

```
Input
    • Distortion values $d_x$ for $x \in \{1 \ldots m\}$
    • Prior $p(x)$ for $x \in \{1 \ldots m\}$
    • Tradeoff parameter $\beta \geq 0$
Sort the words in accordance to their score
    $\beta d_{x_1} + \log(p(x_1)) \leq \cdots \leq \beta d_{x_m} + \log(p(x_m))$
Initialize $k = m$, $a_k = 1$, $p_k = 1$, $\mathcal{J}_k = \beta \sum_x p(x) d_x$,
    $c_{\text{best}} = k$, $\mathcal{J}_{\text{best}} = \mathcal{J}_k$.
Loop: While $k > 0$
    1. Compute $a_{k-1} = a_k - e^{-\beta d_{x_{k-1}}}$
    2. Compute $p_{k-1} = p_k - p(x_{k-1})$
    3. if $p_{k-1} \leq a_{k-1}$
        • Compute $\mathcal{J}_{k-1}$ using (10).
        • If $\mathcal{J}_{k-1} < \mathcal{J}_{\text{best}}$ then set $k_{\text{best}} = k - 1$ and
          $\mathcal{J}_{\text{best}} = \mathcal{J}_{k-1}$.
    4. Set $k \leftarrow k - 1$.
Compute: $q(0|x)$ using (5) and (7)
Output: $q(0|x)$
```

Figure 1. Finding the one class for fixed distortion.

$$H\left[p(x)\right] + \sum_x p(x) \left[q(0|x) \log \left(\frac{p(x) q(0|x)}{q(0)} \right) \right] + \beta \sum_x p(x) q(0|x) d_x$$

The sum can be split according to whether $x \in \mathcal{C}$ and rearranged again:

$$\begin{array}{l} H\left[p(x)\right] + \sum_{x \in \mathcal{C}} p(x) \left[\log\left(p(x)\right) + \beta d_x\right] \\ -\left(\sum_{x \in \mathcal{C}} p(x)\right) \log\left(\sum_{x \in \mathcal{C}} p(x)\right) \\ +\left(\sum_{x \in \mathcal{C}} p(x)\right) \log\left(1 - \sum_{x \notin \mathcal{C}} e^{-\beta d_x}\right) \end{array} . \tag{9}$$

Let $\mathcal{C}_k = \{x_i : 1 \leq i \leq k\}$, $\mathcal{J}_k$ the value of the objective on $\mathcal{C}_k$, $p_k = \sum_{j=1}^{k} p(x_j) = \sum_{x \in \mathcal{C}} p(x)$, and by $a_c = 1 - \sum_{j=k+1}^{m} e^{-\beta d_{x_j}} = 1 - \sum_{x \notin \mathcal{C}} e^{-\beta d_x}$. These quantities can be computed recursively as follows. Let $\mathcal{J}_m = \beta \sum_x p(x) d_x$, $p_m = 1$ and $a_m = 1$. Given $\mathcal{J}_k$, p_k and a_k, we can compute the following in unit time: $a_{k-1} = a_k - e^{-\beta d_{x_k}}$ and $p_{k-1} = p_k - p(x_k)$. Finally, by examining (9) we get,

$$\begin{array}{rcl} \mathcal{J}_{k-1} & = & \mathcal{J}_k - p(x_{k-1})[\beta d_{x_{k-1}} + \log p(x_{k-1})] \\ & & +[p_k \log(p_k) - p_k \log(a_k)] \\ & & -[p_{k-1} \log(p_{k-1}) - p_{k-1} \log(a_{k-1})] \end{array} \tag{10}$$

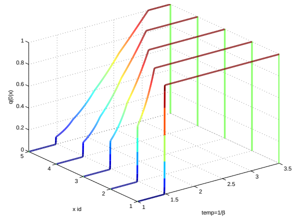

Figure 2. Illustration of the serious of phase transitions of $q(0|x)$ as the temperature $1/\beta$ is modified.

Fig. 1 gives an outline of the algorithm.

The following properties of the solution are worth noting. When the temperature $t = 1/\beta$ is high, all the instances belong to the single cluster with probability 1. As the temperature drops, instances are pulled out of the class, one after the other, as $q(0|x)$ becomes strictly less than 1. Finally, at a critical temperature t_0, all the instances are pulled out of the class, that is, $q(0|x) = 0$ for all x. We show this process in Fig. 2. There are five points, indexed 1 through 5. The distortion of each point is proportional to its index. The y axis is temperature $t = 1/\beta$. For high values of t, all the instances belong to the class with probability 1. At $t \approx 3.1$ the model goes through its first phase transition, as the instance with the highest distortion is pulled out of the class, and its probability of belonging there drops exponentially. There are three more similar phase transition for instances 4, 3 and 2 respectively. Then, at $t \approx 1.5$ the model goes through a discontinuous phase transition and $q(0|x)$ falls to zero for all instances.

In summary, the algorithm iterates between two steps: compute $\mathbf{w}$ given $q(0|x)$ and $q(0)$ using (6) (expectation), and use the algorithm in Fig. 1 to find $q(0|x)$ and $q(0)$ from $d_x = \mathcal{D}\left(\mathbf{v}_x \| \mathbf{w}\right)$ (maximization). In this aspect the algorithm is similar to EM, and thus we might suspect that it has a maximum-likelihood analog (Slonim & Weiss, 2002).

4. Multiclass Clustering

It seems natural to generalize from one class to multiple classes by replacing the one class centroid with $k > 1$ centroids. There are $k + 1$ outcomes for each instance: either code it using one of the k centroids, or code it with itself. We now formalize this extension.

We might at first think that we could just generalize $q(t|x)$ from the previous model to range over a set of $k + m$ values — m points and k centroids — where for given x the value of $q(t|x)$ is non-zero for at most $k + 1$ values of t, the k clusters and self-coding. However, this direct approach leads to a derivation that can not be decomposed nicely as in the one class case, because

$1-q(x|x)$ is not informative about individual clusters, just about their sum.

Instead, we break the coding scheme into two stages. First, given an instance x, we determine whether to code it using one of the centroids or by itself $t = x$. Then, for non-self-coded instances, we decide their cluster. Formally, given x we decide if we want to code it by itself (with probability $q(x|x)$) or by some centroid (with probability $q(0|x)$). Next, if we decide to code the instance with one of the centroids, we denote the centroid's identity by S, and denote the probability of encoding using centroid S given the point identity x and the decision to code it 0 by $r(s|0,x) = \Pr\left[S = s|0,x\right]$ We also define the marginals, $q(s,0) = \sum_x p(x)q(0|x)r(s|0,x)$.

As in (2), we write a rate-distortion objective. The rate equals to the mutual information between possible codings and input variables. The distortion-rate optimization is

$$\min_{\{q(\cdot|x)\},r(s|0,x),\{\mathbf{w}_s\}} \beta \sum_{x,s} p(x)q(0|x)r(s|0,x)\mathcal{D}\left(\mathbf{v}_x\|\mathbf{w}_s\right) + \sum_x p(x)\sum_s q(0|x)r(s|0,x)\log\left(\frac{q(0|x)r(s|0,x)}{q(s,0)}\right) + \sum_x p(x)q(x|x)\log\left(\frac{q(x|x)}{q(x)}\right) \ , \qquad (11)$$

subject to normalization $q(0|x) + q(x|x) = 1$ and $\sum_s r(s|0,x) = 1$. The marginals are defined naturally, $q(x) = p(x)q(x|x)$ and $q(0) = \sum_x p(x)q(0|x)$. Before solving the optimization problem we further assume that every point x is associated with exactly one centroid (if any), that is $r(s|0,x) = 1$ or $r(s|0,x) = 0$. From normalization there is only a single cluster s for which $r(s|0,x) = 1$, denoted by $s(x)$. We also denote by $E(s) = \{x : s(x) = s\}$. We do so for two reasons, first, without doing so we could not separate $q(0|x)$ and $q(s,0)$ from each other (for different values of s) and could not get a solution similar to (5). Second, we show below that we can solve this problem by alternating between two algorithms, one of them is the sequential-IB (sIB) (Slonim, 2003) designed for hard clustering. We call this algorithm MCRD (multiclass rate-distortion-based algorithm).

We now solve the optimization analogously to the derivation starting at (3). After writing the Lagrangian we use its derivations to compute self consistent equations. (details omitted for lack of space). First, we have

$$q(0|x) = \min\left\{\frac{q(s(x),0)}{p(x)}e^{-d_{s(x),x}}, 1\right\} \ , \qquad (12)$$

which is the equivalent of (5). Note that the values of $q(0|x)$ are tied for $x \in E(s)$. Thus, there are k sets of equations, each set tying all points in $E(s)$ and the exact value of $q(0|x)$ for $x \in E(s)$ and $q(s,0)$ can be solved separately using the algorithm of Fig. 1.

Next, we can compute the derivative of the Lagrangian with respect to the other variables and obtain self consistent equations

$$\mathbf{w}_s = \frac{\sum_x p(x)q(0|x)r(s|0,x)\mathbf{v}_x}{\sum_x p(x)q(0|x)r(s|0,x)} = \frac{\sum_{x\in E(s)} p(x)q(0|x)\mathbf{v}_x}{\sum_{x\in E(s)} p(x)q(0|x)} \ ,$$

and $r(s|0,x) \propto q(s,0)e^{-\beta\mathcal{D}(\mathbf{v}_x\|\mathbf{w}_s)}$. The last equation can not be used to solve the problem since we assume that $r(s|0,x)$ is an integer. In practice, we use the following lemma which relates the optimization problem of (11) and the optimization problem of the IB method (Tishby et al., 1999).

Lemma 2 *The following two optimization problems are equivalent up to a linear transformation:*

1. *The optimization problem of* (11) *over* $\mathbf{w}_s$ *and* $r(s|0,x)$*, where we fix* $q(0|x)$ *and* $q(s,0)$*, and* $r(s|0,x) \in \{0,1\}$.
2. *The rate-distortion formulation of the IB method (Slonim, 2003), where the assignment probabilities are either* 0 *or* 1*, and a reweighted prior proportional to* $p(x)q(0|x)$.

(Proof omitted due to lack of space.) Using the lemma and the discussion preceding it, we have an algorithm for MCRD that alternates between two steps: (1) Use the sIB algorithm to set the values of $\mathbf{w}_s$ and $r(s|0,x)$, given $q(0|x)$ and $q(s,0)$, with prior proportional to $p(x)q(0|x)$. (2) Use k calls to the algorithm on Fig. 1 to find $q(0|x)$ and $q(s,0)$ from $d_{s(x),x} = \mathcal{D}\left(\mathbf{v}_x\|\mathbf{w}_s\right)$.

5. Experiments

We compare our algorithm (OCRD-BA) with two previously proposed methods: the IB-related one-class algorithm of Crammer and Chechik (2004) (OC-IB), and a well-known convex optimization method (Tax & Duin, 1999; Schölkopf et al., 2001; Crammer & Singer, 2003) (OC-Convex). We obtained Crammer and Chechik's data and followed their evaluation protocol to achieve comparable results. For lack of space, we just discuss document retrieval experiments, although we obtained qualitatively comparable results on gene expression data as well.

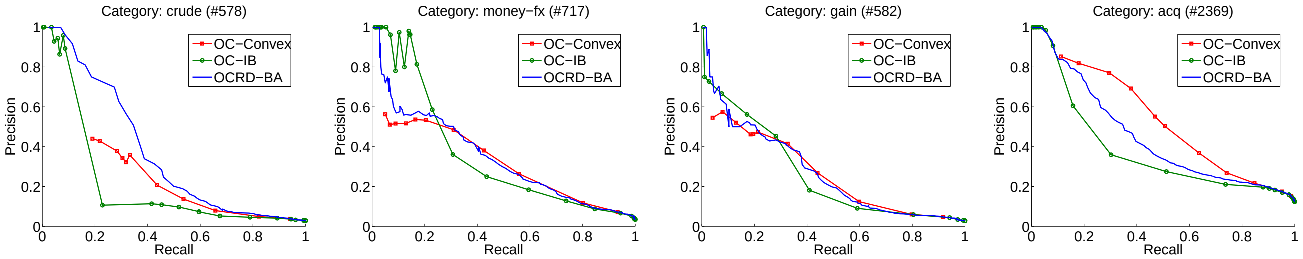

Figure 3. Precision-Recall plots for four (out of five) categories of Reuters-21678 dataset using OC-IB, OC-Convex, and OCRD-BA (this paper).

5.1. Document Retrieval

This is a document retrieval task that was previously described in detail (Crammer & Chechik, 2004, Sec. 6.2). The task uses a subset of the five most frequent categories of Reuters-21578. For each category, half of its documents were used for training, and the remaining half, together with the remaining documents from other categories, were used for evaluation. During training, each of the algorithms searched for a meaningful subset of the training data and generated a centroid. The centroid was used then to label the test data, and to compute recall and precision.

All algorithms used the KL divergence to compare empirical word distributions for different documents. For OC-IB and OC-Convex, we used the parameter values in the previous study (Crammer & Chechik, 2004). For our algorithm, OCRD-BA, we set the prior $p(x)$ to be uniform over the training set, and used a range of values of β that yielded a range of class sizes. We used a single random document to initialize the centroid maintained by OCRD-BA, as was done for OC-IB. We trained five models for each value of β, each using a different random example for initialization, and picked the one which attained the best value of the objective.

After picking a model, we fixed the induced centroid $\mathbf{w}$ and computed the distortion $\mathcal{D}\left(\mathbf{v}_x \| \mathbf{w}\right)$ for all the test examples $\mathbf{v}_x$. We then ran the first half of our algorithm (Fig. 1) to compute the cluster assignments $q(0|x)$. Finally, a test point $\mathbf{v}_x$ was assigned to the class if $q(0|x) = 1$. We used the actual Reuters labels to plot precision and recall values for different β values.

The results are summarized in Fig. 3, where there is one plot per category (except the *earn* category where all algorithms perform the same). As in the previous study (Crammer & Chechik, 2004), we observe that OC-IB achieves better precision than OC-Convex on low recall values. The previous study argues that OC-Convex converges to the center-of-mass of the data for low values of recall while OC-IB exploits local structure and thus performs better. As recall increases, OC-Convex improves and OC-IB degrades, until OC-Convex performs better than OC-IB for high values of recall.

Our method, OCRD-BA, strikes a balance between the two previous methods: at low values of recall, OCRD-BA is comparable in performance to OC-IB and at higher values of recall OCRD-BA is comparable to OC-Convex. Furthermore, in the *crude* category, our method outperformed both algorithms. This suggests that OCRD-BA is similar to OC-IB for small classes and to OC-Convex for large classes. We discuss this issue later.

5.2. Clustering

We evaluated the MCRD algorithm using a synthetic dataset and a real dataset. The synthetic dataset (Synth4G) has 900 points in the plane. Of those, 400 were generated from 4 Gaussian distributions with $\sigma = 0.1$, 100 points from each Gaussian. The remaining 500 point were generated from a uniform distribution. We ran the algorithm allowing up to five clusters with various values of β. The output of the algorithm for four values of β is plotted in Fig. 4. The title of each plots summarize the value of β used, number of points associated with a cluster, and (in parenthesis) the size of each cluster. For low values of β the algorithm prefers to reduce the rate (over distortion) and effectively group all points into a single cluster. As β increases the algorithm uses more clusters until all 5 possible clusters are used (left panel). As β is increased the algorithm prefers to remove points from the clusters, but still use 5 centroids (second panel), until at some point the algorithm only four clusters are used (third panel). Then, for higher values of β five clusters are used again (right panel). This may be due to the fact, that for large β, the actual length scale is small, and thus, practically, there are more than five

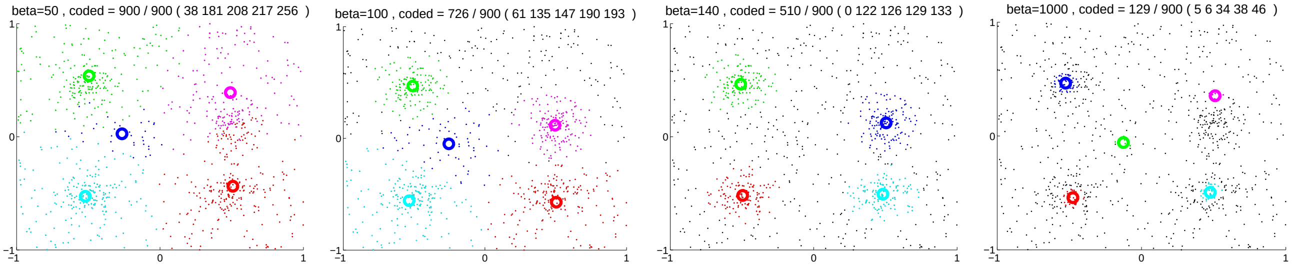

Figure 4. Clusterings produced by MCRD with ($k = 5$) on the synthetic data set for four values of β. Self-coded points are marked by black dots, coded points by colored dots and cluster centroids by bold circles.

clusters (more than five small, dense regions).

We also evaluated on Multi5_1, a real-world high dimensional multiclass dataset which has been used by Slonim et al. (2002) to evaluate the sIB clustering algorithm. The dataset has 500 documents from 5 categories, each represented as a distribution over $2,000$ words. We compare the MCRD algorithm ($\beta = 1.6$) with sIB, which by default, uses all the points in clustering thereby achieving 100% recall. We follow Slonim et al., (2002, Sec. 7.4) to get precision at various recall values for sIB, and for other experimental details. The precision at various recall values is summarized in Fig. 5. We observe that MCRD consistently outperforms sIB at all recall levels. Specifically, MCRD achieves very high precision at low recall values, which is one of the objectives of current work. These experimental results further support our hypothesis that better clustering of the data can be obtained if the algorithm is allowed to selectively leave out data points which are unlikely to help the clustering task.

6. Related Work

Crammer and Chechik (2004) proposed to use the information bottleneck for one-class problems. They compressed the points using two possible events: a point can be either belong to the single class or not. In the former case, the distortion is proportional to the distance between the point and the centroid. In the later case, the distortion equals fixed predefined value R, which intuitively sets the diameter of the class. This formulation suffers from some drawbacks. First, it uses two interacting parameters, the R parameter just discussed, and an inverse temperature β to set the hardness of the solution. In practice, they set β to yield only hard solutions. Second, their distortion does not make sense in term of compression, as the compressor effectively can either approximate a point (using the single class) or ignore it.

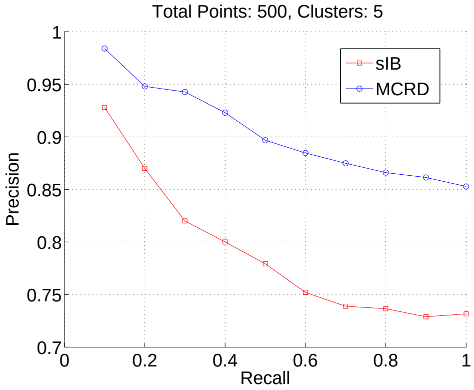

Figure 5. Precision vs. Recall for sIB and MCRD algorithms ($\beta = 1.6$) on the Multi5_1 dataset. Results are obtained by averaging over 5 random permutations of the data. For each permutation, 5 restarts were used and the model with the best objective was selected.

Instead, we use $m+1$ values for the compression variable T, but regularization forces the compressor to generate sparse solutions. In contrast to a fixed nonzero distortion used for out-of-class points, we use a zero distortion because the out-of-class points are not encoded. As a result, our method uses a single inverse temperature parameter to set the size of the class. A point can either be in the class alone (hard assignment) or both in the class and outside (soft assignment).

The centroid is defined as a weighted average of all of the data. Points which belong to the cluster have the same weight, while other points are weighted proportionally to the exponent of their distance from the centroid. This behavior combines properties of two previous approaches. As in Crammer and Chechik's work, the points belonging to the class give an equal contribution in the location of the centroid. But like in discriminative one-class methods (Crammer & Singer, 2003) points outside the class still affect its centroid. Thus our method uses information from all the data, unlike discriminative methods that only see outliers (Tax & Duin, 1999).

The fact that the centroid in our model is contributed to by both typical points and outliers may explain the results of the experiments. The OC-IB method (Crammer & Chechik, 2004) works in a low-recall condition, that is, with a small class. In this condition, points outside the cluster will have a negligible effect on the centroid, yielding the OC-IB solution. For large values of β, points outside the class have a stronger effect on the centroid's location, similarly to discriminative methods (Crammer & Singer, 2003). Furthermore, when using the KL divergence, points that are not contributing to the centroids at all (removed from data), would typically have a divergence of infinity to the centroids. Our methods allows to reduce the effect of outliers (by giving them exponential small weight), but still allow them to contribute to the centroids (positive weight).

Gupta and Ghosh (2006) present an extension of the Crammer and Chechik algorith that clusters points while allowing some of them to be ignored. Recently, Lashkari and Golland (2008) proposed an exemplar-based algorithm , in which any point can serve as a centroid (similarly to k-medians). They show that under some choices, some points can be coded by themselves. Our method is different in that it allows centroids that do not coincide with any data point (similar to k-means).

7. Conclusions

Building on the rate-distortion formulation of the information bottleneck method, we cast the problem of identifying a small coherent subset of data as an optimization problem that trades off class size (compression) for accuracy (distortion). We analyzed a rate-distortion view of the model and demonstrated that it goes through a sequence of phase transitions that correspond to different class sizes. We demonstrated that our method combines the best of two previous methods, each of which is good in a narrower range of class sizes. We also showed that our method allows us to move from one-class to standard clustering, but with background noise left out. The proposed approach for one-class learning can be extended to the idea of regularizing by using constraints over a large set of decisions which can be used for other more complex associations among random variables, and in particular for bi-clustering.

Acknowledgments This material is based in part upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-07-D-0185, and by NSF grant IIS-0513778.

References

Bekkerman, R., & McCallum, A. (2005). Disambiguating web appearances of people in a social network. *WWW*. Chiba, Japan.

Ben-Hur, A., Horn, D., Siegelmann, H., & Vapnik, V. (2001). Support vector clustering. *JMLR*, *2*, 125–137.

Censor, Y., & Zenios, S. (1997). *Parallel optimization: Theory, algorithms, and applications*. Oxford Univ. Press, NY, USA.

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

Crammer, K., & Chechik, G. (2004). A needle in a haystack: Local one-class optimization. *ICML 23*.

Crammer, K., & Singer, Y. (2003). Learning algorithms for enclosing points in bregmanian spheres. *COLT 16*.

Gupta, G., & Ghosh, J. (2006). Bregman bubble clustering: A robust, scalable framework for locating multiple, dense regions in data. *ICDM*.

Lashkari, D., & Golland, P. (2008). Convex clustering with exemplar-based models. *NIPS*.

Schölkopf, B., Burges, C., & Vapnik, V. (1995). Extracting support data for a given task. *KDD 1*.

Schölkopf, B., Platt, J., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation*, *13*, 1443–1472.

Slonim, N. (2003). *The information bottleneck: Theory and applications*. Doctoral dissertation, Hebrew University.

Slonim, N., Friedman, N., & Tishby, N. (2002). Unsupervised document classification using sequential information maximization. *SIGIR*.

Slonim, N., & Weiss, Y. (2002). Maximum likelihood and the information bottleneck. *NIPS*.

Tax, D., & Duin, R. (1999). Data domain description using support vectors. *ESANN* (pp. 251–256).

Tishby, N., Pereira, F., & Bialek, W. (1999). The information bottleneck method. *37th Allerton Conference on Communication, Control, and Computing*. Allerton House, Illinois.