# A Needle in a Haystack: Local One-Class Optimization

**Koby Crammer** KOBICS@CS.HUJI.IL
School of computer science and engineering, the Hebrew University of Jerusalem, Jerusalem 91925, Israel

**Gal Chechik** GAL@ROBOTICS.STANFORD.EDU
Robotics laboratory, Computer science department, Stanford University, Stanford, CA 94305 USA

## Abstract

This paper addresses the problem of finding a small and coherent subset of points in a given data. This problem, sometimes referred to as *one-class* or *set covering*, requires to find a small-radius ball that covers as many data points as possible. It rises naturally in a wide range of applications, from finding gene-modules to extracting documents' topics, where many data points are irrelevant to the task at hand, or in applications where only positive examples are available. Most previous approaches to this problem focus on identifying and discarding a possible set of outliers. In this paper we adopt an opposite approach which directly aims to find a small set of coherently structured regions, by using a loss function that focuses on local properties of the data. We formalize the learning task as an optimization problem using the Information-Bottleneck principle. An algorithm to solve this optimization problem is then derived and analyzed. Experiments on gene expression data and a text document corpus demonstrate the merits of our approach.

## 1. Introduction

The goal of unsupervised learning is to extract concise descriptions of data given empirical samples. However, one is often only interested in modeling small parts of the data and ignore its remaining irrelevant parts. This is for example the case when the data consists of small groups of coherently structured data points, and the rest of the data are irrelevant or mere noise. It is a common scenario in a variety of applications from detecting regions of interest in images to finding subsets of co-expressed genes in genome-wide experiments.

Appearing in *Proceedings of the $21^{st}$ International Conference on Machine Learning*, Banff, Canada, 2004. 

A related problem is demonstrated by the task of information retrieval by search engines. In this application, given a query, a handful of documents is sought out of billions of potential web pages. Unlike classification problems, retrieval systems are often trained only with word phrases or documents marked as relevant, because it is difficult to sample well the space of irrelevant documents. In practice, users tend to consider only the first few pages retrieved by search engines. As a result, a system that retrieves few documents that are all highly relevant, is preferred over a system that retrieves a large collection of documents, many of which are irrelevant, even if the larger collection succeeds to cover more relevant documents. From the point of view of the trade-off between recall (retrieving most relevant documents) vs. precision (retrieving only relevant ones), this application favors high precision values.

In these two general scenarios, one looks for a small but coherent subsets of data items, which can be achieved by finding a small-radius ball that covers as many data points as possible. This problem is known as *one-class classification* (Tax & Duin, 1999), and has two opposite approaches for its solution: Most previous approaches formulate the task as a problem of outlier and novelty detection, in which most of the samples are identified as relevant. Here we take the opposite approach and try to identify a small subset of relevant samples, rather than keep all but few outliers. As we will see below, this "needle in a haystack" approach requires a different formulation of the detection problem.

Current approaches to one-class classification use convex cost functions that focus on large-scale structures in the data: the cost is constant within the ball and grows linearly on its outside (Schölkopf et al. (1995), Tax and Duin (1999) and Ben-Hur et al. (2001) used the Euclidean distance). In a related problem, introduced by Schölkopf et al. (2001), the goal is to separate most of the samples from the origin using a single hyper-plane. Recently, Crammer and Singer (2003) generalized these approaches to the more general family of Bregman Divergences. In all these formulations, due to the convexity of the cost function, the solution converges to the center of mass when the radius of the ball goes to zero, thus ignoring any explicit local structures.

In order to model the fact that the distribution of the points outside the cluster is not of interest, the current paper takes the opposite approach: we use a cost function that grows linearly inside the ball but is kept constant outside it. This cost function is indifferent to the values of the "non-interesting" samples. Flat cost outside the ball is therefore expected to be better than linear cost when the interesting samples are localized in a small region, or when there are relatively few interesting samples. Unfortunately, this cost function leads to a non-convex optimization problem, that is harder to solve exactly. It is therefore necessary to develop approximation algorithms for this problem.

This paper sets out to formalize this problem in the context of an unsupervised learning problem: searching for a compact, yet informative description of the data. We use the *Information Bottleneck* approach (Tishby et al., 1999) (IB) to formalize the learning task as an optimization problem. IB is a general framework that we use here mainly as a tool. It allows us to formalize a stochastic description of the solutions, and to adapt previously studied IB algorithms for the current problem. We use here a previously studied generalization of IB to a rather wide family of distance measures known as Bregman divergences. Since the Euclidean distance is a special case of a Bregman divergence our formulation can also be naturally combined with Mercer kernels, in a manner similar to Schölkopf et al. (1995; 2001).

## 2. Formulation of the Optimization Problem

Assume we are given a set of $m$ samples, $\{\mathbf{v}_x\}_{x=1}^m \in \Lambda^m$, where the identity of a sample is indexed by the random variable $X$ (that is, $x \in \{1, \ldots, m\}$). $\Lambda$ can be a $d$ dimensional space or simplex and $p(x) \stackrel{\text{def}}{=} p(X = x)$ is the prior distribution over the samples. Our goal is to identify a subset of *meaningful* samples from the large set of data points, and we identify the meaningful points by the fact that they are clustered together. The learning task is therefore to find a small body that covers as many samples as possible. To keep the shape of this body simple, we focus here on covering the points with a ball.

We take a probabilistic approach and define $C$ to be the event of being assigned to the ball. This binary event $C \in \{\texttt{TRUE}, \texttt{FALSE}\}$, has a joint distribution with the samples $p(C, x)$, and we denote by $q(C|x)$ the probability that the sample indexed by $x$ is assigned to the ball. This probability can be thought to reflect our belief that the specific sample $\mathbf{v}_x$ is "interesting". We also define the marginal $q(C) = \sum_x q(C|x)p(x)$. Note that we allow *soft assignment* of a sample to the ball, when $q(C|x)$ is neither strictly zero or one. As will be discussed below, our formulation allows to control the level of assignment's softness and to reduce it to the hard limit where the assignment of each sample is dichotomous.

The goal is to find a ball with radius $R$ that is centered at a representative vector $\mathbf{w} \in \Lambda$ such that it covers as many samples as possible. The distance measures that we consider are Bregman divergences, which for the sake of clarity will be defined and discussed in the next section. At this point we treat them as generic distance measures denoted by $B_F(\mathbf{v}_x \| \mathbf{w})$. Clearly, the value of $R$ strongly effects the solution: The smaller the value of $R$ is, less samples will be assigned to the ball. We first treat the case where $R$ is constant and known, and discuss the more general case below. For any value of $R$, a solution to the problem is a set of probabilities $q(C|x)$ for all $x$ and a center of a ball $\mathbf{w}$.

To formalize the learning task as an optimization problem we use the *Information Bottleneck* framework (IB) (Tishby et al., 1999). IB is an elegant formulation for regularized unsupervised learning, that aims to extract a simple, yet meaningful, representation $C$ of a given data $X$. It is usually formalized as an optimization problem that trades off between two terms: One that measures how compact the representation is $I(C; X)$, and another, $I(C; Y)$, that measures how informative it is about an additional given variable $Y$. Both terms are formally quantified using the same functional, the Shannon mutual information $I(X; Y) \stackrel{\text{def}}{=} \sum_{x,y} p(x, y) \log(\frac{p(x,y)}{p(x)p(y)})$.

The current paper uses an alternative, yet mathematically equivalent, formulation of the IB principle. The accuracy of representation is measured here using the average distortion between the data and its compact representation. In the context of the one-class problem, the learning task is written as a minimization of the tradeoff between two terms

$$\min_{q(C|x), \mathbf{w}} \beta \mathcal{D}(C, \mathbf{w}; X) + I(C; X) \tag{1}$$

where $I(C; X)$ is the mutual information between $X$ and $C$, and $\beta$ is a free parameter which determines the tradeoff between the model's accuracy and simplicity.

The first term $\mathcal{D}$ is an average distortion,

$$\begin{aligned} \mathcal{D}(C, \mathbf{w}; X) &= \sum_x p(x) \mathcal{D}(C|x, \mathbf{w}; \mathbf{v}_x) \\ \mathcal{D}(C|x, \mathbf{w}; \mathbf{v}_x) &= q(C|x) B_F(\mathbf{v}_x \| \mathbf{w}) + (1 - q(C|x)) R\,. \end{aligned} \tag{2}$$

For each sample $\mathbf{v}_x$, the distortion term $\mathcal{D}(C|x, \mathbf{w}; \mathbf{v}_x)$ averages terms that correspond to hard assignments of the sample. When $q(C|x) = 1$ ($\mathbf{v}_x$ is certain to be assigned to the ball) the assignment is penalized with a loss equal to the divergence between $\mathbf{v}_x$ and $\mathbf{w}$. When $q(C|x) = 0$ ($\mathbf{v}_x$ is certain not to be assigned to the ball) the assignment is penalized with a constant loss $R$.

The second term of Eq. (1) provides a measure of how strongly the model $C$ compress the data. To compress the structures in $X$ to a simpler representation, we wish to choose a ball that removes information about the specific identity of the points. This becomes clear when rewriting $I(C;X) = h(X) - h(X|C)$ where $h(\cdot)$ is the differential entropy. Since $h(X)$ is a constant, minimizing the information is equivalent to maximizing the entropy $h(X|C)$ which is the fundamental measure of uncertainty about $X$.

To simplify the form of the optimization problem we can rewrite the distortion term of Eq. (2) as $\sum_x p(x) [q(C|x)B_F(\mathbf{v}_x\|\mathbf{w}) + (1 - q(C|x))R] = \sum_x p(x)q(C|x)(B_F(\mathbf{v}_x\|\mathbf{w}) - R) + R\sum_x p(x)$ . The second term is constant because both $R$ and $p(x)$ are given and thus can be omitted without affecting the mathematical properties of the distortion term.

We therefore consider the optimization problem of the combined terms

$$\begin{aligned} \min_{q(C|x),\mathbf{w}} \quad & \mathcal{F}(q(C|x),\mathbf{w}) = I(X;C) \\ & +\beta \sum_x p(x)q(C|x)(B_F(\mathbf{v}_x\|\mathbf{w}) - R) \\ \text{subject to} \quad & q(C|x) \in [0,1] \quad \forall x \end{aligned} \tag{3}$$

Since the objective function contains a product of two variables ($q(C|x)B_F(\mathbf{v}_x\|\mathbf{w})$ , where both $q(C|x)$ and $\mathbf{w}$ are parameters) it is not convex, hence the optimization problem is not convex either.

## 3. Properties of the solution

We now turn to describe the properties of the optimal solution of Eq. (3). We follow the derivation of (Tishby et al., 1999), use some algebraic manipulation, and obtain the following set of self-consistent equations which describe the optimal solution. The first equation describes the marginal over $C$

$$q(C) = \sum_x p(x)q(C|x) \; . \tag{4}$$

The second equation describes the location of the centroid $\mathbf{w}$ in terms of the input samples $\mathbf{v}_x$ and the probabilities $q(C|x)$,

$$\mathbf{w} = \frac{1}{q(C)} \sum_x p(x)q(C|x)\mathbf{v}_x \; . \tag{5}$$

$\mathbf{w}$ is therefore a weighted average of the input samples $\mathbf{v}_x$ weighted by the likelihood probabilities $p(x|C)$. The third equation connects the value of the probabilities $q(C|x)$ to the distance between the centroid $\mathbf{w}$ and each of the points $\mathbf{v}_x$,

$$q(C|x) = 1 / \left\{ 1 + \frac{1 - q(C)}{q(C)} e^{\beta[B_F(\mathbf{v}_x\|\mathbf{w}) - R]} \right\} \; . \tag{6}$$

When $\beta = 0$, the information term in the objective function is dominant, yielding the simplest solution: all the points are assigned to the cluster with the same probability $q(C|x) = q(C)$ regardless of the specific sample value $\mathbf{v}_x$. The specific value of $q(C)$ is determined by ranging over the value of $R$. When $\beta \to \infty$, $q(C|x)$ attains one of three values as follows

$$\lim_{\beta\to\infty} q(C|x) = \begin{cases} 1 & B_F(\mathbf{v}_x\|\mathbf{w}) < R \\ 0 & B_F(\mathbf{v}_x\|\mathbf{w}) > R \\ q(C) & B_F(\mathbf{v}_x\|\mathbf{w}) = R \end{cases} \tag{7}$$

Thus if $\mathbf{v}_x$ is inside (outside) the ball then $q(C|x) = 1$ ($q(C|x) = 0$). If $\mathbf{v}_x$ lies on the ball boundary then $q(C|x) = q(C)$, the a-priori probability of being assigned to the ball. In other words, for a given $\mathbf{w}$, the best assignment for $x$ is to minimize the loss function

$$\mathcal{L} = \min\{B_F(\mathbf{v}_x\|\mathbf{w}) \; , \; R\}.$$

(a) (b)

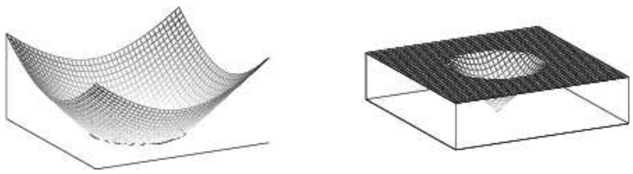

*Figure 1.* Illustration of the loss function in (a) One-class SVM (b) One-class IB. Note the local nature of the loss function in (b).

As discussed above, most previous work on one-class problems (Schölkopf et al., 1995; Tax & Duin, 1999; Schölkopf et al., 2001; Crammer & Singer, 2003) used a convex and unbounded loss functions such as $\max\{0, B_F(\mathbf{v}_x\|\mathbf{w}) - R\}$. This is (up to a constant $R$) the opposite of $\mathcal{L}$. Inside the ball, $\mathcal{L}$ grows linearly, while the other loss function is constant. On the other hand, outside the ball $\mathcal{L}$ is constant while the other loss function grows linearly. An illustration of these two loss functions is given in Fig. 1.

To demonstrate the effect of this difference between the two loss functions, we created a simple synthetic example in which 300 points were normally distributed in two Gaussians centered at $[0.5, 0.9]$ and $[0.9, 0.5]$ and 700 points were uniformly distributed over the 2-d unit square. Applying one-class SVM to this problem tends to find clusters that are centered somewhere between the two Gaussians, because its cost function penalizes for centers that are distant from the center of mass. This result is demonstrated in Fig. 2, where dot-dashed circles correspond to the results of a one-class SVM, each covering a different fraction of the data. One-class IB circles successfully identify one of the Gaussians for a large range of cluster sizes. To simplify the demonstration we consider the points in input space rather than in feature space and did not use any kernel with the one-class IB. The same problem is expected to occur in feature space if kernels are used.

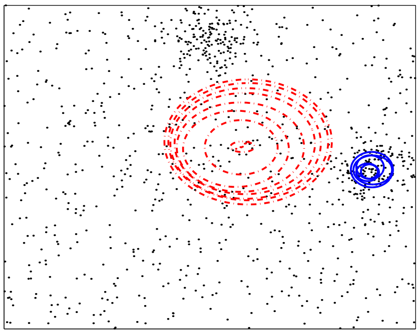

*Figure 2.* Comparison of one-class SVM and IB on a synthetic example. 700 data points are uniformly distributed in the unit square and 300 points are normally distributed in two spheric Gaussians with $\sigma = 1/20$. Each circle is centered at the mean of the cluster, and it size follows 1 standard deviation in both x and y axes. Dot-dashed circles: SVC (Polynomial kernel of degree 2). Solid circles: IB. For one-class IB, runs were repeated 10 times and the result which minimizes the target function was taken.

Until this point, we discussed the case where the radius $R$ was assumed to be known to the algorithm. The effect of $R$ on the solution is crucial: smaller $R$ lead to smaller subsets assigned to the ball. The natural question arises: can this $R$ be set in advance to a "right" value? As in other model-complexity meta parameters (the number of clusters in clustering problems or number of components in PCA) this question is not well defined in an unsupervised learning setting. It is often the case that the data can be described at several resolutions: each revealing different aspects of the data. Which of them is the relevant one depends on the task at hand and is not dictated by the problem. A correct characterization of the data therefore requires to obtain solutions for a spectrum of $R$ values.

The parameter $R$ can in principle be replaced by other global parameters such as the "weight" of the ball. However, we found that such formulations yielded more cumbersome solutions whose properties were more difficult to analyse. In some cases however, it is possible to formally relate $R$ to other global parameters. For example, in the case of an $L_2$ norm, the problem discussed here can be shown to be equivalent to a mixture model of a single Gaussian and a uniform background distribution. In this case $R$ can be explicitly related to the prior weight of the mixture's components. This derivation will be published elsewhere.

## 4. Bregman Divergences

In the original formulation of the IB (Tishby et al., 1999) the input samples represent multinomial distributions over a finite set. As a natural consequence, the definition of the mutual information over the joint distribution gives rise to the Kullback-Leibler (KL) divergence that emerges as a measure of discrepancy between any single sample $\mathbf{v}_x$ and the centroid of a cluster $\mathbf{w}$. In many interesting problems however, representing the data as distributions is not natural. For example, in various experimental measurements the data are the sum of a signal and some near-Gaussian noise, and are well represented as vectors in a Euclidean space. In such cases the Euclidean distance is a more natural divergence. In this paper we follow Crammer and Slonim (2003) which extended the IB to a richer set of possible divergences, called Bregman divergences. A Bregman divergence is defined via a strictly convex function $F : \Lambda \to \mathbb{R}$ defined on a closed, convex set $\Lambda \subseteq \mathbb{R}^n$. $F$ has to satisfy a set of constraints, whose description we omit and refer the reader to Censor and Zenios (1997). All the functions we discuss in this paper obey these constraints and are hence Bregman functions. Assume that $F$ is continuously differentiable at all points of $\Lambda_{int}$, the interior of $\Lambda$, which we assume is nonempty. The Bregman divergence associated with $F$ is defined for $\mathbf{v} \in \Lambda$ and $\mathbf{w} \in \Lambda_{int}$ to be

$$B_F(\mathbf{v}\|\mathbf{w}) \stackrel{\text{def}}{=} F(\mathbf{v}) - [F(\mathbf{w}) + \nabla F(\mathbf{w}) \cdot (\mathbf{v} - \mathbf{w})]\,. \quad (8)$$

Thus, $B_F$ measures the difference between $F$ and its first-order Taylor expansion about $\mathbf{w}$, evaluated at $\mathbf{v}$. The divergences we employ are defined via a single scalar convex function $f$ such that $F(\mathbf{v}) = \sum_{l=1}^{n} f(v_l)$, where $v_l$ is the $l$-th coordinate of $\mathbf{v}$. As a consequence, the Bregman divergences we use are sums of Bregman Divergences per coordinate of the input vectors, $B_F(\mathbf{v}\|\mathbf{w}) = \sum_{l=1}^{n} B_F(v_l\|w_l)$ .

Although Bregman divergences are quite general they share many interesting properties. A property relevant to this paper is stated in the following Lemma.

**Lemma 1 (Convexity of a Bregmanian Ball)** *The set of points* $\{\mathbf{v} \,:\, B_F(\mathbf{v}\|\mathbf{w}) \le R\}$ *is convex .*

The proof of the lemma is straightforward and uses the fact that Bregman divergences are convex in their first argument. Thus, although we use a rich family of divergences we are still guaranteed to have only convex bodies. A straightforward consequence of the lemma is that if some point $\mathbf{v}$ belongs to the ball around $\mathbf{w}$ then all the points constituting the line connecting $\mathbf{v}$ and $\mathbf{w}$ also belong to it.

Bregman distances provide a generalization over several commonly used distance measures. In this paper we demonstrate our algorithms and their analysis with two commonly used divergences. The first is the square distance between $\mathbf{v}$ and $\mathbf{w}$, $B_F(\mathbf{v}\|\mathbf{w}) = \frac{1}{2}\|\mathbf{v} - \mathbf{w}\|^2$, obtained by setting $f(x) = (1/2)\,x^2$. In this case $\Lambda \subset \mathbb{R}^n$. The second divergence can be obtained when $\Lambda$ is the $n$-dimensional simplex. Setting $f(v) = v\log(v)$ yields the KL divergence $B_F(\mathbf{v}\|\mathbf{w}) = \sum_{l=1}^{n} v_l \log\left(\frac{v_l}{w_l}\right)$. Two other Bregman divergences are Itakura-Saito (Censor &

**Input:** Set of Points $\{\mathbf{v}_x\}_{x=1}^m$ ; Divergence $B_F$ ; Radius $R > 0$

**Initialize:**

- Pick a point $\mathbf{v}_{\tilde{x}}$ at random
- Set $\mathbf{w} = \mathbf{v}_{\tilde{x}}$, $\tilde{q}(C|x) = 1$, $q(C|x) = 0$ for $x \neq \tilde{x}$

**Loop:** For $t = 1, 2, \ldots, T$

- Draw a random permutation $\pi$ of $1 \ldots m$
- For $i = 1, 2, \ldots m$
  1. Set $x = \pi(i)$
  2. If $q(C|x) = 1$ remove $\mathbf{v}_x$ from the cluster:
     $$\begin{array}{lll} \mathbf{w} & \leftarrow & \frac{q(C)\mathbf{w} - p(x)\mathbf{v}_x}{q(C) - p(x)} \\ q(C) & \leftarrow & q(C) - p(x) \\ q(C|x) & \leftarrow & 0 \end{array}$$
  3. Set $\pi_{\mathbf{v}_x} = \frac{p(x)}{q(C)+p(x)}$ , $\pi_c = \frac{q(C)}{q(C)+p(x)}$
  4. Set $\tilde{\mathbf{w}} = \pi_{\mathbf{v}_x}\mathbf{v}_x + \pi_c \mathbf{w}$
  5. If $\pi_c B_F(\mathbf{w}\|\tilde{\mathbf{w}}) + \pi_{\mathbf{v}_{\tilde{x}}} B_F(\mathbf{v}_{\tilde{x}}\|\tilde{\mathbf{w}}) < \pi_{\mathbf{v}_{\tilde{x}}} R$ then merge the sample with the current cluster: $\mathbf{w} \leftarrow \tilde{\mathbf{w}}$ ; $q(C) \leftarrow q(C) + p(x)$ ; $q(C|x) \leftarrow 1$

**Return:** Centroid $\mathbf{w}$ ; Assignment's indicators $q(C|x)$

*Figure 3.* The sequential algorithm for *one-class IB*

Zenios, 1997) ($f(x) = -\log(x)$ , $\Lambda \subset \mathbb{R}_+^n$) designed for speech analysis and Unnormalized Relative Entropy (Censor & Zenios, 1997) ($f(x) = x\log(x) - x$ , $\Lambda \subset \mathbb{R}_+^n$) often used with data represented by counts (positive integers). The Itakura-Saito is not convex in its second argument.

## 5. Algorithms

In this section we describe an algorithm that finds a local optimum for the problem defined in Eq. (3). Its output is the center of the ball $\mathbf{w}$ and the probabilistic assignments $q(C|x)$ of points to the ball.

Several algorithms were developed for the original IB problem (see Slonim (2002) for review and comparison). Some of these algorithms can be adapted to the problem discussed in this paper, but some may not be easily extended to general Bregman divergences. For example, the iterative algorithm by Tishby et al. (1999) is based on iterations between the three self-consistent equations of the bottleneck solution Eqs. (4,5,6). Unfortunately, for general Bregman divergences, it is no longer guaranteed that $B_F$ is convex with respect to its second argument, therefore the optimization over Eq. (5) may not find its minimum.

Among the algorithms developed for IB, we chose to adapt the sequential algorithm (Slonim, 2002) because it is fast, easy to implement, and usually finds good local minima. It was designed for the hard clustering case, in which the assignment of samples to clusters is deterministic rather than stochastic. While hard solutions may be inferior to soft solutions, they are often more easily interpretable, since the set of samples that belong to a cluster is clearly defined. An additional advantage of hard assignments is that the first term of the objective function can be further simplified, $I(C;X) = h(C) - h(C|X) = h(C)$, since for hard assignment $q(C|x) \in \{0,1\}$ and $h(C|x) = 0$.

The sequential algorithm operates in iterative steps. At each step, the algorithm picks a sample $\mathbf{v}_x$ and tests if modifying its status *and updating the centroid accordingly* would decrease the value of the objective function. That is, for a sample that is not assigned to the ball, the algorithm checks if assigning it to the ball decreases the objective function. Similarly, the algorithm checks if excluding a sample that is already assigned to the ball decreases the objective function. We now describe in details the first case, the derivation of the second case is similar.

Assume we have a set of parameters $q(C|x) = \sum_x p(x)q(C|x)$ , $\mathbf{w} = \frac{\sum_x p(x)q(C|x)\mathbf{v}_x}{\sum_x p(x)q(C|x)}$ and let us focus on some specific $\tilde{x}$ for which $q(C|\tilde{x}) = 0$. Let $\tilde{q}(C|x)$ be the probability of assignment after making the change, and check how the objective function $\mathcal{F}$ changes by setting $\tilde{q}(C|\tilde{x}) = 1$. We thus set $\tilde{q}(C|x) = q(C|x)$ for all $x \neq \tilde{x}$ and $\tilde{q}(C|\tilde{x}) = 1 \neq 0 = q(C|\tilde{x})$. Let us denote by $\tilde{q}(C|x), \tilde{q}(C), \tilde{\mathbf{w}}$ the set of parameters after inserting the sample $\tilde{x}$ into the set of interesting samples. It is straightforward to verify that $\tilde{q}(C) = q(C) + \tilde{p}(x)$, and

$$\tilde{\mathbf{w}} = \frac{q(C)\mathbf{w} + \tilde{p}(x)\mathbf{v}_{\tilde{x}}}{\tilde{q}(C)} = \frac{q(C)\mathbf{w} + \tilde{p}(x)\mathbf{v}_{\tilde{x}}}{q(C) + \tilde{p}(x)} . \quad (9)$$

Let us now compute the difference in the objective function for both assignments of parameters. From the equality $I(C;X) = h(C)$ above we obtain that the difference between the two compression term is a difference between two values of the entropy functional, $\Delta h = h(\tilde{C}) - h(C)$. For the simplicity of presentation we omit this term, and discuss only the case for which $\beta^{-1} = 0$. It is straightforward to compute these terms for finite $\beta$ as well :

$$\begin{aligned} &\mathcal{F}\{\tilde{q}(C|x), \tilde{\mathbf{w}}\} - \mathcal{F}\{q(C|x), \mathbf{w}\} \quad (10) \\ &\quad = \sum_x p(x)q(C|x)\left[B_F(\mathbf{v}_x\|\tilde{\mathbf{w}}) - B_F(\mathbf{v}_x\|\mathbf{w})\right] \\ &\qquad + p(\tilde{x})\left(B_F(\mathbf{v}_{\tilde{x}}\|\tilde{\mathbf{w}}) - R\right) . \end{aligned}$$

where the equality stems from $0 = q(C|x) \neq \tilde{q}(C|x) = 1$. We now use the definition of Bregman divergences (Eq. (8)) together with Eq. (9) and have the first term,

$$\begin{aligned} &\sum_x p(x)q(C|x)\left[B_F(\mathbf{v}_x\|\tilde{\mathbf{w}}) - B_F(\mathbf{v}_x\|\mathbf{w})\right] \quad (11) \\ &\quad = q(C)B_F(\mathbf{w}\|\tilde{\mathbf{w}}) . \end{aligned}$$

Finally, plugging Eq. (11) into Eq. (10) we obtain $\mathcal{F}\{\tilde{q}(C|x), \tilde{\mathbf{w}}\} - \mathcal{F}\{q(C|x), \mathbf{w}\} = q(C)B_F(\mathbf{w}\|\tilde{\mathbf{w}}) +$

$p(\tilde{x})\left(B_F\left(\mathbf{v}_{\tilde{x}} \| \tilde{\mathbf{w}}\right) - R\right)$. By defining the following Bernoulli distributions $\pi_{\mathbf{v}_{\tilde{x}}} = \frac{p(\tilde{x})}{q(C)+p(\tilde{x})}$ and $\pi_c = \frac{q(C)}{q(C)+p(\tilde{x})}$, we have that the sample $\mathbf{v}_{\tilde{x}}$ is inserted to the cluster of interesting points if,

$$\pi_c B_F\left(\mathbf{w} \| \tilde{\mathbf{w}}\right) + \pi_{\mathbf{v}_{\tilde{x}}} B_F\left(\mathbf{v}_{\tilde{x}} \| \tilde{\mathbf{w}}\right) < \pi_{\mathbf{v}_{\tilde{x}}} R \; . \qquad (12)$$

Plugging Eq. (9) into the left hand side of Eq. (12),

$$\pi_c B_F\left(\mathbf{w} \| \pi_c \mathbf{w} + \pi_{\mathbf{v}_{\tilde{x}}} \mathbf{v}_{\tilde{x}}\right) + \pi_{\mathbf{v}_{\tilde{x}}} B_F\left(\mathbf{v}_{\tilde{x}} \| \pi_c \mathbf{w} + \pi_{\mathbf{v}_{\tilde{x}}} \mathbf{v}_{\tilde{x}}\right) \qquad (13)$$

This term equals to the weighted average of two distortion terms, between $\mathbf{w}$ and $\mathbf{v}_{\tilde{x}}$ to their common average, where the two distortions are weighted in according to $\pi_c$ and $\pi_{\mathbf{v}_{\tilde{x}}}$. We thus call Eq. (13) the *Average Bregman Divergence (ABD)* [1]. Finally, using again the special form of Bregman divergences we can write Eq. (13) as,

$$\pi_{\mathbf{v}_{\tilde{x}}} F(\mathbf{v}_{\tilde{x}}) + \pi_c F(\mathbf{w}) - F\left(\pi_{\mathbf{v}_{\tilde{x}}} F(\mathbf{v}_{\tilde{x}}) + \pi_c F(\mathbf{w})\right) \; . \qquad (14)$$

Eq. (14) enables us to compute the Average Bregman Divergence in a very simple manner. To conclude this part of the algorithm, given a new sample which is not part of the cluster, we evaluate Eq. (14). If it is smaller than $\pi_{\mathbf{v}_{\tilde{x}}} R$, we merge it into the cluster, otherwise we ignore it. The case where we need to decide if to exclude a point out of the cluster is similar. In fact, we can always remove the given point $\mathbf{v}_x$ out of the cluster and then apply the above procedure to determine if we like to include it back or not. We also like to note in passing that in the case of the Euclidean distance the condition of Eq. (12) can be written in a simpler form, $\frac{1}{2}\|\mathbf{v}_{\tilde{x}} - \mathbf{w}\| < R\frac{q(C)+p(\tilde{x})}{q(C)}$. Thus, in the case of the Euclidean distance we check whether the sample $\mathbf{v}_{\tilde{x}}$ lies in a ball around the current center $\mathbf{w}$. Where the radius of the ball depends also on the weight ($q(C)$) of the current cluster and the weight of the specific sample ($p(\tilde{x})$). Finally, for the general case $\beta < \infty$, the following condition determines if the current example $\mathbf{v}_{\tilde{x}}$ should be merged into the cluster or not, $\pi_c B_F\left(\mathbf{w} \| \pi_c \mathbf{w} + \pi_{\mathbf{v}_{\tilde{x}}} \mathbf{v}_{\tilde{x}}\right) + \pi_{\mathbf{v}_{\tilde{x}}} B_F\left(\mathbf{v}_{\tilde{x}} \| \pi_c \mathbf{w} + \pi_{\mathbf{v}_{\tilde{x}}} \mathbf{v}_{\tilde{x}}\right) + \frac{h(\tilde{C})-h(C)}{\beta(q(C)+p(\tilde{x}))} < \pi_{\mathbf{v}_{\tilde{x}}} R$. An outline of the algorithm is given in Fig. 3.

## 6. Experiments

We evaluate the performance of our approach on two real-life and high dimensional problems: identifying predictive genes, and document retrieval.

### 6.1. Gene expression in B-Cell Lymphoma

We applied our approach to the problem of finding a small sets of interesting genes in micro arrays gene expression experiments. In genome-wide gene expression experiments, a large set of genes is pre-defined and their level of expression in various tissues is measured. The goal is often to identify subsets of genes that can be related to a biological process or function. In some cases labeled sets of tissues are available, and the relevant genes can be found using supervised techniques. In other cases, labeling is unknown. An important example is the case of identifying subtypes of disease, where gene expression are used to uncover different biological mechanisms that lead to similar clinical symptoms. Within this framework we have chosen to use the data of (Alizadeh & et al., 2000), that contains gene expression levels in tissues of B-cells lymphoma patients. It was previously used in numerous (mostly supervised) machine learning studies. Importantly, there also exists data on the survival of 39 patients, thus predicting this survival rate provides an external independent measure of performance for any automated approach. Using their expertise, Alizadeh and colleagues were able to identify genes that are believed to separate B cell lymphoma into two subtypes, which also differ considerably in the expected survival rate. In the context of the current paper, Our goal with this data is to identify a single cluster that contains genes that are good predictors of survival in a fully automated unsupervised manner.

The data consists of the expression profiles of $4,026$ genes over $46$ tissues. To reduce the dimensionality of the data to a level that can be handled by the SVM package we used (OSU SVM, www.eleceng.ohio-state.edu/~maj/osu_svm), we first chose the $500$ genes with the highest single-gene information $I(x) = D_{KL}[p(y|x)||p(y)]$ (where $p(x,y) = 1/[1+\exp(data(x,y))]$ is the probability

[1] When the Bregman divergence is the Kullback-Leibler, Eq. (13) becomes equals to the symmetric KL or Jensen-Shannon divergence $JS_\pi(\mathbf{v},\mathbf{w}) = \pi KL(\mathbf{v}|\pi\mathbf{v} + (1-\pi)\mathbf{w}) + (1-\pi)KL(\mathbf{w}|\pi\mathbf{v} + (1-\pi)\mathbf{w})$.

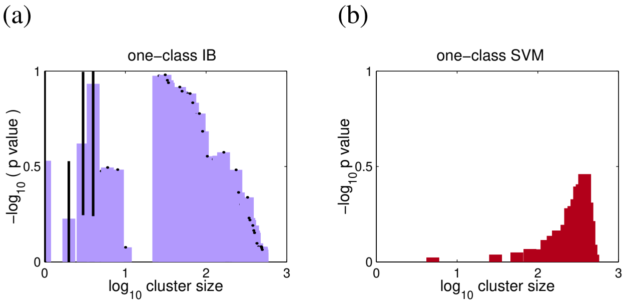


*Figure 4.* Performance of one class IB and SVM on predicting survival of B-cell lymphoma patients based on gene expression profile from a single coherent class. y-axis is $\log_{10}$ of the $p$-value obtained from using all genes in the cluster for predicting survival rate using linear regression. (a) one class IB with $\beta^{-1} = 0$. (b) one class SVM polynomial kernel of degree 2. Similar results obtained also with RBF kernel at a large range of Gamma values. Error bars denote standard deviation across all clusters of the same size. Since the size of the clusters in one-class IB is an outcome of the algorithm, rather than determined externally, not all cluster sizes provide a good solution.

assigned with a gene $x$ and a tissue $y$). We then applied both one-class-IB and one-class-SVM to the genes, each gene being a vector in $\mathbb{R}^{46}$. For one-class-SVM, we enumerated over $\nu$ (the fraction of outliers) thus obtaining an optimal cluster for each cluster size. For one-class IB, we enumerated over the cost $R$, yielding optimal clusters of different sizes. For each cost, optimization was repeated 100 times with different random seeds, and the result with best target function value was used. All the genes of the chosen clusters, were then used in a linear regression to predict the survival rate. The log of the $p$ values of this prediction are plot in Fig. 4 as a function of the cluster size. Since for one-class-IB several cost values may lead to the same cluster size, we plot the mean and standard deviation of $\log(p \text{ value})$ over all costs that yield the same cluster size.

The performance of one-class-IB in predicting survival rates largely outperforms that of one-class-SVM for almost all cluster sizes, and is in particular better than one-class-SVM for smaller clusters. We believe that the reason is that small SVM clusters are strongly biased to the center of the whole data, thus are not sensitive enough to local structures.

### 6.2. Document Retrieval

We further examined the usefulness of our approach on another real life complex data, by evaluating its performance in a documents retrieval problem. We used the Reuters-21578 corpus (available at www.daviddlewis.com/resources/testcollections), which contains $10,789$ documents, each associated with categories taken from a set of about $90$ topics. We used the ModApte pre-processing of the corpus and the feature selection schema described in Slonim (2002), and picked a dictionary of size $2,000$ .

In our experiments we used a subset of the data that contained the five most frequent categories: *earn* (3964 documents), *acq* (2369), *money-fx* (717), *grain* (582) and *crude* (578). Finally, we represented each document as a multinomial distribution over term counts. For each of the above five topics we repeated the following experimental setup. The data was split into a training set that contained half of the documents from the chosen topic, and a test set that contained all other documents. For example, in the case of the *earn* category, the training contained $1,982$ documents (all from the *earn* category), and the remaining $8,807$ documents constituted the test set ($1,982$ document from the *earn* category and $6,825$ documents from other categories).

We implemented two algorithms. First, one-class-IB, as described in Sec. 5 (OC-IB for short) and second, the one-class algorithm described in (Crammer & Singer, 2003), called here OC-Convex. Both algorithms used the

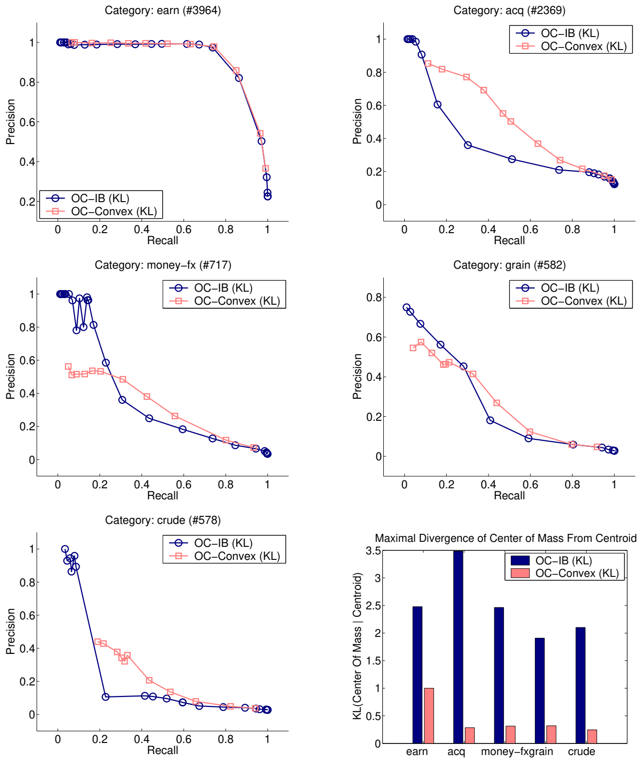


*Figure 5.* Plot of Precision vs. Recall for the five topics of Reuters-21578. Two algorithms are compared: one-class trained with convex loss function (OC-Convex) and the one-class IB (OC-IB). The number near the name of the category in the title indicates the total number of examples associated with this category. Right-bottom-figure: plot of the maximal distance of the resulting centroid and the center of mass of the training set. The maximum is taken for each of the algorithms over the possible values of tuning parameters.

Kullback-Leibler divergence. OC-Convex uses a single parameter $\nu$ which controls the number of outliers, that was set here to range between $0.04\%$ and $99.999\%$. OC-IB is tuned by setting the constant-cost parameter $R$. To cover the range of parameters values we first sampled pairs of samples from the training set, and used it to estimated the maximal and minimal divergence. We generated a list of possible costs approximately between this two estimates. For every possible cost value we trained the sequential algorithm $10$ times and picked the most populated ball.

We evaluated both learning algorithms as following. For both algorithms we set the value of the controlling parameter and ran the algorithm which generated a ball parameterized by a center and a radius. For each of the documents in the test set, we tested if it falls inside or outside this ball, and computed the two following measures (using the test set only). The *recall* which equals to the fraction of documents labeled *earn* that are contained in the ball from the total number of documents labeled *earn*. The second measure is the *precision* which equals to the fraction of documents labeled *earn* that are contained in the ball from the total number of documents which are contained in the ball. In both algorithms the control parameter enables the user to trade between recall and precision. Recall-Precision curves are given in Fig. 5.

The result over the five topics share a common behavior. For large values of recall the one-class combined with the convex loss achieves slightly better precision than OC-IB. As we decrease the recall value, the gap between the value of the precision decreases until the recall attains some value (around $20\%$), in which the performance of the one-class IB is better than the one-class with convex loss. For very low values of recall the one-class IB clearly outperforms the one-class with convex loss. In fact, in three different categories, the ball obtained by the one-class algorithm with the convex loss did not contain even a single document from the test set. This result is in accordance with our intuition stated above: For very low values of recall the one-class with convex loss convergence to the center of mass of the input examples, regardless of any other property of the input data. On the other hand, the one-class IB locates areas of the input data which is dense related to its neighborhood. This intuition is further supported by the bottom-right panel in Fig. 5. For each of the categories and algorithms we computed the divergence of the centroid from the center of mass of examples. The height of the bar designates the maximal divergence over the possible set of control parameters for each of the algorithms.

## 7. Summary and Conclusions

We addressed the problem of finding a *small* coherent subsets of data points in a high dimensional complex data. We demonstrated why current approaches to the problem are less sensitive to local structures of the data when searching for small subsets, and described a localized cost function that improves this sensitivity. Building on the elegant Information-Bottleneck approach we cast the learning task as an optimization problem which trades-off between the two opposite demands of simplicity and accuracy. We derived a simple algorithm that is guaranteed to converge to a local minima that also works well in practice. In two real-world domains, when searching for small clusters, our approach provides an improvement over current one-class methods which are based on the principle of large margin. Our approach also provides a well-defined probability measure which indicates for each of the input examples the probability that it is belonging to the single cluster or not.

Similarly to previous approaches (Crammer & Singer, 2003) our framework can be combined with a general notion of divergence measures - the Bregman divergences. Furthermore, Since the Euclidean norm is a Bregman divergence, we can also combine Mercer kernels (Schölkopf et al., 1995) with our method.

The current paper focused on training a *single* one-class model for the input data. However, real world complex data, often contains several distinct regions that can be learned separately. It is therefore an interesting question how current algorithms for combining several one-class classifiers can be combined with our approach.

Several extensions of the current work are of interest. First, it will be interesting to explore alternative macroscopic parameters that control the problems solutions. Specifically, the constant cost $R$ may be replaced with another parameter which will control the size of the cluster $q(C)$ similar to the parameter $\nu$ in previous formulations. Another extension is to use our approach to solve problems of set covering in the context of information theory and rate distortion.

**Acknowledgments:** We thank Amir Globerson and Yoram Singer for fruitful discussions and careful reading of the manuscript.

## References


Alizadeh, A., & et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene-expression profiling. *Nature*, *4051*, 503–511.

Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. (2001). Support vector clustering. *J. of Mach. Learn. Res.*, *2*, 125–137.

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

Crammer, K., & Singer, Y. (2003). Learning algorithms for enclosing points in bregmanian spheres. *Proceedings of the Sixteenth Annual Conference on Computational Learning Theory*.

Crammer, K., & Slonim, N. (2003). Bregman information bottleneck. Presentation at the Workshop on Information Bottleneck and Information Distortion, Neural Information Processing Systems, Vancouver, Canada.

Itakura, F., & Saito, S. (1970). A statistical method for estimation of speech spectral density and formant frequencies. *Electronics & Communications in Japan*, *53*, 36–43.

Schölkopf, B., Burges, C., & Vapnik, V. (1995). Extracting support data for a given task. *First International Conference on Knowledge Discovery & Data Mining (KDD)*. AAAI Press.

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. (2002). *The information bottleneck: Theory and applications*. Doctoral dissertation, The Hebrew University.

Tax, D., & Duin, R. (1999). Data domain description using support vectors. *Proceedings of the European Symposium on Artificial Neural Networks* (pp. 251–256).

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