# Bregman Bubble Clustering: A Robust, Scalable Framework for Locating Multiple, Dense Regions in Data

Gunjan Gupta Joydeep Ghosh

Department of Electrical & Computer Engineering
University of Texas at Austin, Austin, TX 78712, USA.
{ggupta/ghosh}@ece.utexas.edu

## Abstract

*In traditional clustering, every data point is assigned to at least one cluster. On the other extreme, One Class Clustering algorithms proposed recently identify a single dense cluster and consider the rest of the data as irrelevant. However, in many problems, the relevant data forms multiple natural clusters. In this paper, we introduce the notion of Bregman bubbles and propose Bregman Bubble Clustering (BBC) that seeks $k$ dense Bregman bubbles in the data. We also present a corresponding generative model, Soft BBC, and show several connections with Bregman Clustering, and with a One Class Clustering algorithm. Empirical results on various datasets show the effectiveness of our method.*

## 1 Introduction

Many unsupervised learning problems involve summarizing the data using a small number of parameters. Algorithms such as K-Means partition the data into $k$ clusters directly for a given $k$, while other methods give a hierarchy of clusters. However, in many real-world problems, only a subset can be summarized well, while the rest of the data shows little or no clustering tendencies. Typically in such cases only a portion of the data, containing multiple natural groupings, is relevant. These include: (1) Market-basket data, where only a subset of customers show coherent behavior. (2) Many web-mining applications where recovering the most relevant items of key categories is more important than obtaining an exhaustive list. (3) Many types of bioinformatics datasets. For example, gene-expression datasets measure expression level of genes compared to a control across a few thousand genes. The experiments typically cover only a specific "theme" such as stress-response, and therefore only a few genes related to the conditions show good clustering. Biologists are interested in recovering small, multiple clusters formed from a small subset of genes that show strongly correlated expression patterns [1]. Other types of biological data that share similar properties include protein mass spectroscopy and phylogenetic profile data.

For such situations, we would like to design clustering algorithms that are (1) scalable, (2) can cluster only a variable fraction of the whole dataset, (3) find multiple clusters, and (4) can work with a wide variety of distance measures. Existing density-based methods for finding dense clusters such as DBSCAN [5] are not suitable for many such situations because of implicit metric assumptions, and are not scalable to very large problems since they either require an in-memory $O(n^2)$ distance matrix, or an efficient index that usually does not exist for high-dimensional datasets. Recently introduced One Class Clustering methods such as OC-IB [3] and BBOCC [8] use local search that are much more scalable and general [2] but can only find a single dense cluster.

## 2 Contributions

This paper substantially generalizes the single-cluster approach of BBOCC, and consists of three major extensions/enhancements that lead to a robust and scalable framework for finding multiple dense clusters. Our main contributions are as follows:

1. We present a generalization of BBOCC called Bregman Bubble Clustering (BBC) that can simultaneously find $k$ dense clusters. BBC with a time and space complexity of $O(nd)$ (for a dataset with $n$ data points in $d$ dimensions) for each iteration, is scalable

[1] Often such clusters map to biological processes that are involved in the specific context, for example stress.

[2] Both methods use *Bregman divergences*, a large class of divergence measures that includes Squared Euclidean distance, K-L divergence, Itakura-Saito distance and Mahalanobis distance.

to much larger and higher-dimensional datasets than existing density-based methods. It also extends Euclidean distance centric density-based clustering to a large class of popular divergences known as Bregman divergences.

2. We present an extension to BBC called *Pressurization* that substantially improves the quality of the local search and overcomes local minima while preserving the scalability of the local search approach. The resulting clustering is extremely robust and shows very low sensitivity to initialization.

3. We develop a generative (soft) model consisting of a mixture of $k$ exponentials and a uniform "background" distribution that leads to several insights into the problem of finding dense clusters using Bregman divergences. BBC and many existing clustering algorithms are shown to be special cases of this model.

4. We performed evaluations on a variety of datasets showing the effectiveness of our framework on low, medium and very high-dimensional problems, as compared to Bregman Clustering, Single Link Agglomerative and DBSCAN.

5. An appropriate model selection strategy is discussed.

**Notation**: Bold faced variables, e.g. $\mathbf{x}$ represent vectors. Sets are represented by calligraphic upper-case alphabets such as $\mathcal{X}$ and are enumerated as $\{\mathbf{x}_i\}_{i=1}^n$ where $\mathbf{x}_i$ are the individual elements. $|\mathcal{X}|$ represents the size of set $\mathcal{X}$. Capital letters such as $X$ are random variables. $\mathbb{R}$ and $\mathbb{R}^d$ represent the domain of real numbers and a $d$-dimensional vector space respectively.

## 3 Related Work

A variety of density-based methods are based on the idea of local density estimation to cluster a part of the data. These approaches also have the ability to find arbitrary shaped clusters. DBSCAN [5] is a popular method in the database community for clustering and indexing 2-D and 3-D datasets. Jiang et al. [11] applied density based clustering to gene-expression data.

A One Class Clustering algorithm [3] called OC-IB was proposed [3] that uses the notion of a Bregmanian ball to find a single dense region using an iterative relocation algorithm. Gupta and Ghosh [8] described an improved local search called BBOCC, and provide performance guarantee using an enumeration-based seeding. Earlier approaches to One Class Clustering [18, 17] used convex cost functions that are good for finding large-scale structures, or correspondingly, for finding a small number of outliers. Crammer and Chechik [3] explain why these approaches are not suitable when the goal is to find locally dense regions. Our approach is similar to that of [8] and [3], with the additional property that we can find multiple dense regions.

In the context of clustering microarray data, discovering overlapping gene clusters is popular since many genes participate in multiple biological processes. Gene Shaving [9] uses PCA to find a small subset of genes that show strong expression change compared to the control sample, and allows them to be in multiple clusters.

## 4 Bregman Bubble Clustering (BBC)

### 4.1 Bregman Divergences

*Bregman divergences* form a family of distance measures, defined as follows: Let $\phi : S \mapsto \mathbb{R}$ be a strictly convex function defined on a convex set $S \subseteq \mathbb{R}^d$, such that $\phi$ is differentiable on $int(S)$, the interior of $S$. The Bregman divergence $D_\phi : S \times int(S) \mapsto [0, \inf)$ is defined as $D_\phi(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - (\mathbf{x} - \mathbf{y}, \bigtriangledown\phi(\mathbf{y}))$ where $\bigtriangledown\phi$ is the gradient of $\phi$. For example, for $\phi(\mathbf{x}) = \parallel \mathbf{x} \parallel^2$, $D_\phi(\mathbf{x}, \mathbf{y}) = \parallel \mathbf{x} - \mathbf{y} \parallel^2$, which is the Squared Euclidean Distance. Similarly, other forms of $\phi$ lead to other Bregman divergences such as Logistic Loss, Itakura-Saito Distance, Hinge Loss, Mahalanobis Distance and KL Divergence [15, 2].

### 4.2 Cost Function

Let $\mathcal{X} = \{\mathbf{x}\}_{i=1}^n \subset C \subseteq \mathbb{R}^d$ be the set of data points. Let $\mathcal{G} \subset \mathcal{X}$ represent a non-exhaustive clustering consisting of $k$ clusters $\{\mathcal{C}_j\}_{j=1}^k$ with $\mathcal{X} \backslash \mathcal{G}$ points that are "don't care", i.e., they do not belong to any cluster. For a given Bregman Divergence $D_\phi(\mathbf{x}, \mathbf{y}) \mapsto [0, \infty)$, and a set of $k$ cluster representatives $\{\mathbf{c_j}\}_{j=1}^k \in \mathbb{R}^d$ for the $k$ clusters in clustering $\mathcal{G} = \{\mathcal{C}_j\}_{j=1}^k$, we define the cost $Q$ as the average distance of all points in $\mathcal{G}$ from their assigned cluster representative:

$$Q(\mathcal{G}, \{\mathbf{c_j}\}_{j=1}^k) = \frac{1}{|\mathcal{G}|} \sum_{j=1}^{k} \sum_{i:\mathbf{x}_i \in \mathcal{C}_j}^{|\mathcal{C}_j|} D_\phi(\mathbf{x}_i, \mathbf{c}_j), \qquad (1)$$

### 4.3 Problem Definition

Given $s$, $k$ and $D_\phi$ as inputs, where $s$ out of $n$ points from $\mathcal{X}$ are to be clustered into a clustering $\mathcal{G} \subset \mathcal{X}$ consisting of $k$ clusters, where $1 \leq k < n$ and $k \leq s \leq n$, we define the clustering problem as:

**Definition 1**: *Find the smallest cost $\mathcal{G}$ consisting of $k$ clusters in $\mathcal{X}$, such that $|\mathcal{G}| = s$.*

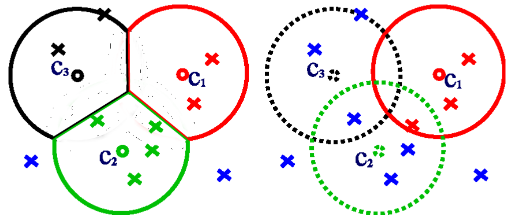
[3] Also known as One Class Classification.

## 4.4 Bregman Bubbles

A *Bregmanian ball* [3] $B_\phi(r, \mathbf{c})$ with radius $r$ and centroid $\mathbf{c}$ defines a volume in $\mathbb{R}^d$ such that all points $\mathbf{x}$ where $D_\phi(\mathbf{x}, \mathbf{c}) \leq r$ are enclosed by the ball. Given a set $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ of $n$ points in $\mathbb{R}^d$, the cost of the ball is defined as the average $D_\phi(\mathbf{x}, \mathbf{c})$ of all points enclosed by it.

Given a set of $k$ cluster representatives, and a fixed $s$, it can be shown that the clustering that minimizes $Q$ consists of: (1) the assignment phase, where each point is assigned to the nearest cluster representative, and (2) picking points closest to their representatives first until $s$ points are picked. Let $r_{max}$ represent the distance of the last ($s^{th}$) picked point from its cluster representative.

These clusters can be viewed as $k$ *Bregman bubbles* such that: (1) they are either pure Bregmanian balls of radius $r \leq r_{max}$ or are (2) *touching* bubbles that form when two or more Bregmanian balls, each of radius $r_{max}$ overlap. Two Bregmanian balls $B_\phi(\mathbf{c1}, r_1)$ and $B_\phi(\mathbf{c2}, r_2)$ are said to overlap when $\exists \mathbf{x} : (D_\phi(\mathbf{x}, \mathbf{c1}) < r_1) \wedge (D_\phi(\mathbf{x}, \mathbf{c2}) < r_2)$. At the point of contact, the touching bubbles form linear boundaries[4] that result from assigning points to the closest cluster representative. For the part of its boundary where a bubble does not touch any other bubble, it traces the contour of a Bregmanian ball of radius $r_{max}$. Therefore, bubbles arise naturally as the optimum solution for $Q$ for a given $s$, $k$ and $D_\phi$.



(a) (b)

**Figure 1. An illustration showing (a) three Bregman bubbles, and (b) a Bregmanian ball (solid line), and two other possible balls (dotted lines). The union of the points enclosed by the three possible balls in (b) is the same as the set of points enclosed by the three bubbles.**

Figure 1 illustrates a 2-D example of Bregman bubbles vs. balls. Unlike Bregmanian balls, the boundary of the Bregman bubbles can only be defined in the context of other bubbles touching it. It is important to note that the volume of the convex hull of points in one bubble could be smaller than that of the adjacent touching bubble, and the bubbles could also have different number of points assigned to them.

**Algorithm 1** BBC-S

```
Input: Set $\mathcal{X} = \{\mathbf{x}\}_{i=1}^n \subset C \subseteq \mathbb{R}^d$, Bregman divergence $D_\phi$,
    no. of clusters $k$, desired clustering size $s$.
Output: Partitioning $\mathcal{G}^*$ containing $k$ clusters$\{\mathcal{C}_j\}_{j=1}^k$, and the
    corresponding $k$ cluster representatives $\{\mathbf{c_j^*}\}_{j=1}^k$.
Method:
    if $\{\mathbf{c_j}\}_{j=1}^k = \varnothing$ then
5:      Initialize cluster representatives: $\{\mathbf{c_j}\}_{j=1}^k \in C$
    end if
    $\mathcal{G}^l = \varnothing; \mathcal{G} = \varnothing; q = \infty; q_p = \infty;$
    repeat
        for $i = 1$ to $n$ do
10:         $[d_i^{min}, lab_i] = \min_{j=1}^k (D_\phi(\mathbf{x_i}, \mathbf{c_j}))$
        end for
        $[val, idx] = sort(\mathbf{d}^{min})$
        $q^{tmp} = 0; s^c = 0; \{\mathcal{C}_j\}_{j=1}^k = \varnothing$
        while $(s^c < s)$ do
15:         $s^c = s^c + 1;$
            $q^{tmp} = q^{tmp} + val(s^c)$
            Add $\mathbf{x_{idx(s^c)}}$ to cluster $\mathcal{C}_{lab(idx(s^c))}$
        end while
        $\{\mathbf{c_j^P}\}_{j=1}^k$= $\{\mathbf{c_j}\}_{j=1}^k$
20:     $q^p = q; q = q^{tmp}/s$
        $\mathcal{G}^l = \mathcal{G}; \mathcal{G} = \{\mathcal{C}_j\}_{j=1}^k$
        for $j = 1$ to $k$ do
            $\mathbf{c_j} = \frac{1}{|\mathcal{C}_j|} \sum_{i:\mathbf{x}_i \in \mathcal{C}_j}^{|\mathcal{C}_j|} \mathbf{x}_i$
        end for
25: until $(\mathcal{G}^l = \mathcal{G}) \wedge q_p = q$
    Return $\{\mathbf{c_j^*}\}_{j=1}^k = \{\mathbf{c_j}\}_{j=1}^k; \mathcal{G}^* = \mathcal{G}$
```

## 4.5 BBC-S Algorithm

For most real life problems, even for a small $s$, finding the globally optimal solution for problem definition 1 would be too slow. However, a fast iterative relocation algorithm that guarantees a local minima exists. *Bregman Bubble Clustering-S* (BBC-S, Algorithm 1) starts with $k$ centers and a size $s$ as input. Conceptually, it consists of three stages: (1) the assignment phase, where each point is assigned to the nearest cluster representative, (2) picking points closest to their representatives first until $s$ points are picked, and (3) updating the centers. It is interesting to note that stages 1 and 3 of BBC-S are identical to the Assignment Step and the Re-estimation step of the Bregman Hard Clustering [2], properties that lead to the unification described in Section 8. Stages 1, 2 and 3 are repeated until there is no change in assignment between two iterations - i.e. the algorithm converges. Algorithm 1 describes a more detailed implementation of BBC-S where line number 10 represents Stage 1, lines 14 to 18 map to Stage 2, while lines 22-24 represent Stage 3. We randomly pick $k$ data points from $\mathcal{X}$ as the starting cluster representatives, but alternative initialization schemes could be implemented.

[4] This can be shown to be true for all Bregman divergences [2].

**Theorem 4.1.** **[2]:** *Let $X$ be a random variable taking values in $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset C \subseteq \mathbb{R}^d$ following $\nu$* [5]*. Given a Bregman divergence $D_\phi : C \times int(C) \mapsto [0, \inf)$, the problem*

$$\min_{c \in C} E_\nu[D_\phi(X, c)]$$

*has a unique minimizer given by $\mathbf{c}^* = \mu = E_\nu[X]$.*

**Proposition 4.2.** *Algorithm 1 is guaranteed to converge to a local minima for all Bregman divergences.*

This follows from the observation that at each iteration the cost $Q$ either declines or stays the same. It is easy to show that for a given set of cluster representatives, the cluster assignment stages 1 and 2 give the lowest possible cost. Therefore, in stages 1 and 2, the cost cannot increase but can decrease. If no point's cluster assignment changes in stages 1 and 2, the cost stays the same and the algorithm converges. Similarly the cost $Q$ at stage 3 can either decline or stay the same because of Theorem 4.1. By using a heap sort at stage 2, each iteration of BBC-S takes $O(nkd + s \log n)$ time making it really fast.

### 4.6 BBC-Q: Dual formulation with fixed $q_{max}$

An alternative formulation of the BBC algorithm is possible where a threshold cost $q_{max}$ is specified rather than the size $s$:

**Definition 2***: Find the largest $\mathcal{G}$ consisting of $k$ clusters in $\mathcal{X}$ with cost no more than $q_{max}$.*

We can show that this definition also results in Bregman bubbles as the optimal solution for a given set of $k$ cluster representatives. Definitions 1 and 2 are equivalent, since for a given $q_{max}$ there exists a largest $s$ for $k$ bubbles, and for the same $s$, the same solution has the same smallest possible cost $q_{max}$. Algorithm 1 can be easily modified to work with $q_{max}$ by modifying Stage (2) to stop adding points when the cost is more than $q_{max}$. However, this seemingly minor modification results in two very different algorithms. For a fixed $s$ as input, for iterations in sparse regions the bubbles expand until $s$ points are covered. As the bubbles move into denser regions, their radii shrink. BBC-Q does not have this property and generally gives worse performance when the bubbles are small. This observation led us to the idea of Pressurization discussed in Section 6. Furthermore, in many problems there is no intuitive way to determine $q_{max}$, while users often have an idea of what fraction of their data might cluster well. This makes **Definition 1** a more natural choice.

## 5 Soft BBC

### 5.1 Bregman Soft Clustering

Banerjee et al. [2] proposed a soft clustering algorithm called *Bregman Soft Clustering* as a mixture model consisting of $k$ distributions, taken from the family of *regular exponential distributions* (that include well known distributions such as Gaussians, Multinomials, Poisson, etc.). They went on to prove the following important result:

**Theorem 5.1.** *There is a bijection between regular exponential families and regular Bregman divergences (equation 2).*

$$p_{(\psi,\theta)}(\mathbf{x_s}) = exp(-\beta D_\phi(\mathbf{x_s}, \mu)) f_\phi(\mathbf{x_s}) \qquad (2)$$

where $\phi$ is a convex function, and the conjugate function of $\psi$, $D_\phi$ is the corresponding Bregman divergence, $p_{(\psi,\theta)}$ is the corresponding regular exponential distribution with cumulant $\psi$, $f_\phi$ is a uniquely determined normalizing function that depends on the choice of $\phi$, $\beta$ is a scaling factor, $\mu$ is the expectation parameter, $\theta$ are the natural parameters of $p^\phi$, and $\mathbf{x_s}$ is the sufficient statistics vector corresponding to $\mathbf{x}$. For the sake of notational simplicity, for the rest of the paper, unless stated explicitly otherwise, when we mention $\mathbf{x}$ we implicitly refer to the sufficient statistics of $\mathbf{x}$, i.e. $\mathbf{x_s}$.

Examples of Bregman divergences and the corresponding exponential distribution that have been popular for both hard and soft clustering models include squared Euclidean Distance (Gaussian distribution), KL-divergence (multinomial distribution) and Itakura-Saito distance.

### 5.2 Motivation for Soft BBC

BBC can be thought of as a non-exhaustive hard clustering where points can belong to either one of the $k$ clusters or to a "don't care" group. Correspondingly, Soft BBC can be formulated as modeling the data as a mixture of $k$ distributions from the exponential family and an additional "background" distribution that corresponding to the "don't care" points. Since we are trying to find $k$ dense clusters, for a good solution the "don't care" group should be the least dense. One way to model this low density background is with a uniform distribution. The goal of building such a Soft BBC model is to give us deeper insights into the implicit modeling assumptions behind BBC.

### 5.3 Model

The Soft BBC model is defined as follows: Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ be the dataset consisting of $n$ i.i.d. points and $k$ be the desired number of clusters. Let $\mathcal{Y} = \{Y_i\}_{i=1}^n$ be the hidden random variables taking values from 0 to $k$ corresponding to $k + 1$ mixture components associated with the data points, where 0 corresponds to a uniform background distribution, and 1 to $k$ corresponds to $k$ exponential mixtures. The likelihood of the data points is given by:

[5] Theorem 4.1 is more general in that it holds for any measure $\nu$ defined on the samples. For the BBC formulation we assume all points to have the same weight, but we later discuss Soft BBC in Section 5 that uses a probabilistic weighting.

```
Algorithm 2 Soft BBC
Input: Set $\mathcal{X} = \{\mathbf{x}\}_{i=1}^n \subset C \subseteq \mathbb{R}^d$, Bregman divergence $D_\phi$, no.
    of clusters $k$, $p_0$, specifying the background distribution, $\alpha_0$ for
    Case B.
Output: $\Theta^*$, local maximizer of $L(\Theta|\mathcal{X})$ (equation 4) where $\Theta =$
    $\{\{\theta_j, \alpha_j\}_{j=1}^k, \alpha_0\}$ for case (A) and $\{\theta_j, \alpha_j\}_{j=1}^k$ for case (B),
    soft partitioning $\{\{p(Y_i = j|\mathbf{x}_i)\}_{j=0}^k\}_{i=1}^n$.
Method:
    Initialize $p_0$, $\{\theta_j, \alpha_j\}_{j=1}^k$ with some $0 \leq p_0 < 1$, $\theta_j \in C$,
    $\alpha_j \geq 0$, such that $\sum_{j=0}^k \alpha_j = 1$.
    repeat
      {The E Step}
      for $i = 1$ to $n$ do
        for $j = 0$ to $k$ do
          $p(Y_i = j|\mathbf{x}_i)$ is computed from equation (7) and (8),
          where $p_{(\psi,\theta)}(\mathbf{x}_i|\theta_j)$ is defined by equation 2.
        end for
      end for
      {The M Step}
      for $j = 0$ to $k$ do
        Update $\alpha_j$ using equation 10 for case A and 13 for case B.
        Update $\theta_j$ using equation 12.
      end for
    until convergence
```

$$p(\mathbf{x_i}) = \sum_{j=1}^{k} \alpha_j p_{(\psi,\theta)}(\mathbf{x}_i|\theta_j) + \alpha_0 p_0, [i]_1^n \tag{3}$$

where $\{\alpha_j\}_{j=0}^k$ denotes the distribution priors, $\{p_{(\psi,\theta)}(\cdot|\theta_j)\}_{j=1}^k$ the conditional distributions of the $k$ clusters, and $p_0$ denotes the probability density of the uniform distribution. Assuming the points are sampled i.i.d., the log-likelihood of the observed data is given by:

$$L(\Theta|\mathcal{X}) = \sum_{i=1}^{n} \log(\sum_{j=1}^{k} \alpha_j p_{(\psi,\theta)}(\mathbf{x}_i|\theta_j) + \alpha_0 p_0) \tag{4}$$

where $\Theta$ denotes the priors and mixture component parameters. It is non-trivial to directly optimize the likelihood function due to the presence of mixture components.

### 5.4 Soft BBC EM Algorithm

Since $p_0$ is a uniform distribution by definition, $1/p_0$ defines the volume of its domain. This domain should include the convex hull of $\mathcal{X}$, which yields an upper bound for $p_0$. In equation 4, keeping all other parameters constant, a lower value of $p_0$ will always result in a lower likelihood. For now, we only consider the case where $p_0$ is set to a fixed value. Therefore, the only parameters we can optimize over are the priors $\{\alpha_j\}_{j=0}^k$ and the exponential mixture parameters $\{\theta_j\}_{j=1}^k$. We consider two slightly different scenarios: (A) where $\alpha_0$ is a variable parameter, and (B) where $\alpha_0$ is a fixed value $\leq 1$. To maximize the log-likelihood function, we adopt a standard EM-based approach [14] and first construct the negative free energy function:

$$\begin{aligned} F(\tilde{P}, \Theta) = & \sum_{i=1}^{n} E_{\tilde{p}(Y_i,\mathbf{x}_i)}[\log p(\mathbf{x}_i, Y_i|\Theta)] \\ & - \sum_{i=1}^{n} E_{\tilde{p}(Y_i,\mathbf{x}_i)}[\log p(Y_i|\mathbf{x}_i)] \end{aligned} \tag{5}$$

where $\tilde{P} = \{\{\tilde{p}(Y_i = j|\mathbf{x}_i)\}_{i=1}^n\}_{j=1}^k$ are the current estimates of $\mathcal{Y}$. It can be shown that the EM procedure with the **E** and **M** steps alternately optimizing $F(\tilde{P}, \Theta)$ over $\tilde{P}$ and $\Theta$ is guaranteed to converge to a local maxima $\tilde{P}^*$ and $\Theta^*$. Furthermore, it can be shown that a local maxima of $F(\tilde{P}, \Theta)$ leads to a local maxima on the original likelihood given by equation 4. Hence we will now focus on obtaining the updates involved in the **E** and **M** steps for the two cases.

**Case (A): $\alpha_0$ is not fixed**

**E-Step**: In this step we optimize $F(\tilde{P}, \Theta)$ (equation 5) over $\tilde{P}$ under the constraints that the $\sum_{j=0}^k \tilde{p}(Y_i = j|\mathbf{x}_i) = 1, [i]_1^n$, and $\tilde{p}(Y_i = j|\mathbf{x}_i) >= 0, \forall i, j$. Using Lagrange multipliers $\{\lambda_i\}_{i=1}^n$ for the $n$ equality constraints and taking derivatives w.r.t. $\tilde{p}(Y_i = j|\mathbf{x}_i)$, we obtain the update equation for re-estimating the probability of each point coming from any of the 0 to $k$ components, given the current model parameters:

$$\log p(\mathbf{x}_i, Y_i = j|\Theta) - 1 - \log \tilde{p}(Y_i = j|\mathbf{x}_i) - \lambda_i = 0 \tag{6}$$

where $p(\mathbf{x}_i, Y_i = j|\Theta)$ is $\alpha_j p_{(\psi,\theta)}(\mathbf{x}_i|\theta_j)$ for $1 \leq j \leq k$ and $\alpha_0 p_0$ for $j = 0$. On eliminating the Lagrange multipliers, we obtain:

$$\tilde{p}(Y_i = j|\mathbf{x}_i)^* = \frac{\alpha_j p_{(\psi,\theta)}(\mathbf{x_i}|\theta_j)}{\sum_{j=1}^k \alpha_j p_{(\psi,\theta)}(\mathbf{x}_i|\theta_j) + \alpha_0 p_0}, 1 \leq j \leq k \tag{7}$$

$$= \frac{\alpha_0 p_0}{\sum_{j=1}^k \alpha_j p_{(\psi,\theta)}(\mathbf{x}_i|\theta_j) + \alpha_0 p_0}, j = 0 \tag{8}$$

**M-Step**: In this step we optimize $F(\tilde{P}, \Theta)$ over $\Theta$ under constraints $\sum_{j=0}^k \alpha_j = 1$ and $\alpha_j \geq 0, \forall j$. It can be shown that the inequality constraints are not binding. Using Lagrange multiplier $\zeta$ for the constraint and taking derivatives w.r.t. $\alpha_j$, $[j]_0^k$, we obtain:

$$\sum_{i=1}^{n} \frac{\tilde{p}(Y_i = j|\mathbf{x}_i)}{\alpha_j} + \zeta = 0, [j]_0^k \tag{9}$$

and on eliminating $\zeta$, we obtain:

$$\alpha_j^* = \frac{\sum_{i=1}^n \tilde{p}(Y_i = j|\mathbf{x}_i)}{n}, [j]_0^k \tag{10}$$

Note that the update equation for the background distribution prior, $\alpha_0$, turns out to be the same as that for the exponential mixture distributions $\alpha_1$ to $\alpha_k$. The optimal mixture component parameter estimation can be obtained by setting derivatives over $\{\theta_j\}_{j=1}^{n}$ to 0 as follows:

$$\sum_{i=1}^{n} \tilde{p}(Y_i = j|\mathbf{x_i}) \nabla_{\theta_j} p_{(\psi,\theta)}(\mathbf{x}_i|\theta_j) = 0 \tag{11}$$

This results in the update equation for the exponential distribution mixtures $\{\theta\}_{j=1}^{k}$ as the weighted average of $\mathbf{x}$ [2]:

$$\theta_j = \frac{\sum_{i=1}^{n} p(Y_i = j|\mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{n} p(Y_i = j|\mathbf{x}_i)} \tag{12}$$

**Case (B): $\alpha_0$ is fixed**

**E-Step**: Since keeping $\alpha_0$ fixed does not result in any additional constraints, this step is identical to that of case A.

**M-Step**: Keeping $\alpha_0$ constant modifies the constraints on the priors so that we now require $\sum_{j=1}^{k} \alpha_j = 1 - \alpha_0$ and $\alpha_j \geq 0, \forall j$. As before, the inequality constraints are not binding and by using a Lagrange multiplier and taking derivatives, we arrive at:

$$\alpha_j^* = (1 - \alpha_0) \frac{\sum_{i=1}^{n} \tilde{p}(Y_i = j|\mathbf{x}_i)}{\sum_{j=1}^{k} \sum_{i=1}^{n} \tilde{p}(Y_i = j|\mathbf{x}_i)} \tag{13}$$

The optimal mixture component parameters are obtained exactly as in case A.

**Choosing an appropriate $p_0$**: For case (A) of the Soft BBC algorithm, one can show that the parameter $\alpha_0$ is essentially a function of $p_0$ given by the relation (from the $M$ step):

$$\alpha_0 = \frac{1}{n} \sum_{i=1}^{n} \frac{\alpha_0 p_0}{\sum_{j=1}^{k} \alpha_j p_{(\psi,\theta)}(\mathbf{x}_i|\theta_j) + \alpha_0 p_0} \tag{14}$$

Using this relation, for a given $\alpha_0$ and a set of mixture component parameters, it is possible to solve for $p_0$. But one cannot do this in the EM framework since the best value for $p_0$ is always the highest possible one. However this relationship allows us to calculate the value of $p_0$ for the initial seed parameters. A fast approximation of $p_0$ can be estimated by (1) performing the first E step (equations 7 and 8), then (2) computing the $p_{max}^{i} = \max_{j=0}^{k}(p(Y_i = j|\mathbf{x}_i))$ for each $\mathbf{x}_i$, and then (3) picking $p_0$ as the $s^{th}$ largest value in $p_{max}^{i}[i]_1^n$ where $s = \lceil \alpha_0 n \rceil$.

## 6 Improving local search with Pressurization

BBC-S is able to find locally dense regions because of its ability to explicitly ignore large amounts of data by considering only points close to the cluster representatives for cluster membership. During each iteration, the bubble representatives move to a lower cost nearby location. But when the dense bubbles are naturally small, i.e. threshold $s$ is small, only a few close neighbors get assigned, thereby decreasing the mobility of the representatives at each iteration. This makes it difficult for BBC-S to find small, dense regions far from initial seed locations. On the other hand, starting with a large $s$ would be contrary to the goal of finding small dense regions. This problem is even more severe with BBC-Q, since the bubbles cannot expand automatically in sparser regions.

Is there a way to improve upon the ability of BBC-S to "expand" in a sparse region, while still optimizing clustering over small, dense regions? We start by defining a concept called *Bregman bubble pressure* that is analogous to the pressure around air bubbles in a body of water on Earth. When air-bubbles rise in a column of water, the outside pressure drops, and the bubbles expand. In the case of BBC-S, we can imagine this external pressure as being inversely proportional to the input threshold $s$; a larger threshold corresponds to a smaller external pressure, leading to larger bubbles.

**BBC-Press**: We propose an algorithmic enhancement to BBC called *Pressurization* that is designed to improve the quality of the local minima discovered. We start the first iteration of BBC-S with a small enough pressure to cause all points to be assigned to some cluster, and slowly increase the pressure after each iteration. An additional parameter $\gamma \in [0, 1)$ that controls the rate of pressure increase is used as an exponential decay parameter, and $s_j = s + \lfloor (n - s)\gamma^{j-1} \rfloor$ is used instead of $s$ for the $j^{th}$ iteration. Convergence is tested only after $(n - s)\gamma^{j-1} < 1$. A somewhat slower but more robust alternative involves running BBC-S to full convergence after each recomputation of $s_j$, and yields slightly better empirical results. Pressurization can also be implemented for BBC-Q by varying $q_{max}$ instead of $s$.

**Soft BBC-Press**: Pressurization can also be extended to Soft BBC for Case B when $\alpha_0$ is not updated. When $\alpha_0$ and $p_0$ are large (close to 1), only a small amount of data is "explained" by the $k$ exponential mixtures. This may lead to bad local minima problems similar to (although less severe than) the one faced in BBC. Therefore, we propose a soft version of Pressurization that takes a decay parameter $\tau \in [0, 1)$ and runs Soft BBC (Case B) multiple times as follows: (1) start with some initial model parameters $\{\theta_j^1\}_{j=1}^{k}$ and run Soft BBC to convergence. (2) at trial $r$ set $\alpha_0$ to $\alpha_r = \alpha_0(1 - \tau^{r-1})$, and for $r > 1$ set current model parameters to the output of last trial: $\{\theta_j^r\}_{j=1}^{k} = \{\theta_j^{r-1}\}_{j=1}^{k}$. Repeat step (2) until $\alpha_r - \alpha_0$ is smaller than $\epsilon$, and then perform a final run with $\alpha_r = \alpha_0$.

## 7 Extension to Pearson Distance

*Pearson Correlation* ($P$) is a popular similarity measure for clustering gene-expression and other biological datasets. Pearson Distance ($D_P$) between two data points $\mathbf{x}$ and $\mathbf{y}$ is defined as $1 - P(\mathbf{x}, \mathbf{y})$, and is also equal to the Squared Euclidean Distance between z-scored [6] $\mathbf{x}$ and z-scored $\mathbf{y}$. When $D_\phi$ is replaced by $D_P$ in Equation 1, we refer to $Q$ as *Average Pearson Distance*(APD). The following directly follows from a proof given by Dhillon and Modha [4]:

**Proposition 7.1.** *For any cluster $\mathcal{C}_j$ in $\mathcal{G}$, the cluster representative $\mathbf{c_j}^*$ that minimizes contribution to APD by that cluster is equal to the mean vector of the points in $\mathcal{C}_j$ projected onto a sphere of unit radius, i.e. $\mathbf{c_j}^* = \underset{\mathbf{c_j}}{argmin}(APD(\mathcal{C}_j, \mathbf{c_j})) = \frac{\mathcal{C}_j^m}{\|\mathcal{C}_j^m\|}$, where $\mathcal{C}_j^m = \frac{1}{|\mathcal{C}_j|} \sum_{i:\mathbf{x}_i \in \mathcal{C}_j}^{|\mathcal{C}_i|} zscore(\mathbf{x}_i)$.*

Therefore, when $D_\phi$ is replaced with $D_P$ for BBC-S (Algorithm 1), the optimal representative for each cluster is computed by averaging the z-scored points rather than the original points, and then again z-scoring the resultant mean. This minor modification makes BBC-S applicable to $D_P$ guaranteeing a local minima in terms of APD cost [7].

## 8 A unified framework

### 8.1 Unifying Soft BBC & BBC

We are now ready to look at how the generative model Soft BBC relates to the BBC problem, specifically the formulation where the number of points classified into the $k$ real clusters (excluding the "don't-care" cluster) is fixed (**Definition 1**, Section 4.3), and show the following:

**Proposition 8.1.** *BBC optimizes a lower bound on the log-likelihood objective function of Soft BBC.*

*Proof.* Let us consider the cost function:

$$L_2(\Theta|\mathcal{X}) = \sum_{i=1}^{n} E_{p^\dagger(Y_i=j|\mathbf{x}_i,\Theta)}[\log p(\mathbf{x}_i, Y_i = j|\theta_j)] \tag{15}$$

where $p^\dagger(Y_i = j|\mathbf{x}_i, \Theta) = 1$ for $j = \underset{0 \le j \le k}{\text{argmax}}\, p(\mathbf{x}_i, Y_i = j|\theta_j)$ and 0 otherwise, which is essentially equivalent to the posterior class probabilities based on the hard assignments used in BBC. It can be shown [12] that for a fixed set of mixture parameters $\Theta = \{\theta\}_{j=1}^k$, and $L(\Theta|\mathcal{X})$ being the log-likelihood objective of Soft BBC (Equation 4):

$$L_2(\Theta|\mathcal{X}) \le L(\Theta|\mathcal{X}) \tag{16}$$

This result is independent of the choice of priors $\{\alpha_j\}_{j=0}^k$. Note that while $L(\cdot)$ depends upon the priors while $L_2(\cdot)$ does not. For our choice of mixture components, based on Equations 2 and 16, one can readily obtain the following form for $L_2(\cdot)$:

$$L_2(\Theta|\mathcal{X}) = \sum_{j=1}^{k} \sum_{\forall Y_i = j}^{k} \log p^\phi(\mathbf{x}_i) - \tag{17}$$

$$\beta D_\phi(\mathbf{x}_i, \theta_j) + \sum_{\forall Y_i = 0} \log(p_0)[i]_{i=1}^n$$

If the number of points assigned to the uniform distribution is fixed to $n - s$, $s$ points are assigned to the $k$ exponential distributions, and $p_0$ and $\beta$ are fixed, we can see from Equation 17 that:

**Proposition 8.2.** *Maximizing $L_2(\Theta|\mathcal{X})$ is identical to minimizing the BBC objective function $Q$ (Equation 1).*

From Proposition 8.2 and Equation 16 we have the proof for Proposition 8.1. □

**Proposition 8.3.** *BBC with a fixed $s$ as input (Definition 1, Section 4.3) is a special case of Soft BBC with fixed $\alpha_0$.*

*Proof.* Let us consider an extreme case when $\beta \to \infty$ for Soft BBC (see Equation 4 and 2). Then the class posterior probabilities in Soft BBC converge to hard assignment (BBC) ensuring that $L(\Theta|\mathcal{X}) = L_2(\Theta|\mathcal{X})$ in Equation 17. Since BBC is equivalent to optimizing $L_2(\Theta|\mathcal{X})$ (Proposition 8.2), we can also view BBC with fixed $s$ as input as a special case of Soft BBC with fixed $\alpha_0$. □

### 8.2 Other unifications

The following other interesting unifications can also be shown easily for our framework:

1. BBC is a special case of BBC-Press when $\gamma = 0$.

2. Bregman Bubble Clustering becomes BBOCC when $k$=1.

3. Soft BBC[8] reduces to Bregman Soft Clustering when $p_0 = 0$.

4. Bregman Bubble Clustering reduces to Bregman Hard Clustering (which is a special case of Bregman Soft Clustering) when $q_{max} = \infty$ (for BBC-Q) or when $s = n$ (for BBC-S).

Figure 2 summarizes the hierarchy of algorithms descending from BBC-Press and Soft BBC. We could think of BBC as a search under "constant pressure", and for Bregman Hard Clustering this pressure is zero. Note that for $k = 1$, Bregman Clustering is not very meaningful [9],

[6] Often used in statistics, normally performed between points across a dimension. Here we perform it between dimensions for each data point.

[7] The same modification works for BBC-Q also.

[8] For both cases A and B.

[9] For $k = 1$, Bregman Soft Clustering returns a single exponential distribution fit to the whole data while Bregman Hard Clustering simply returns the mean of the whole data.

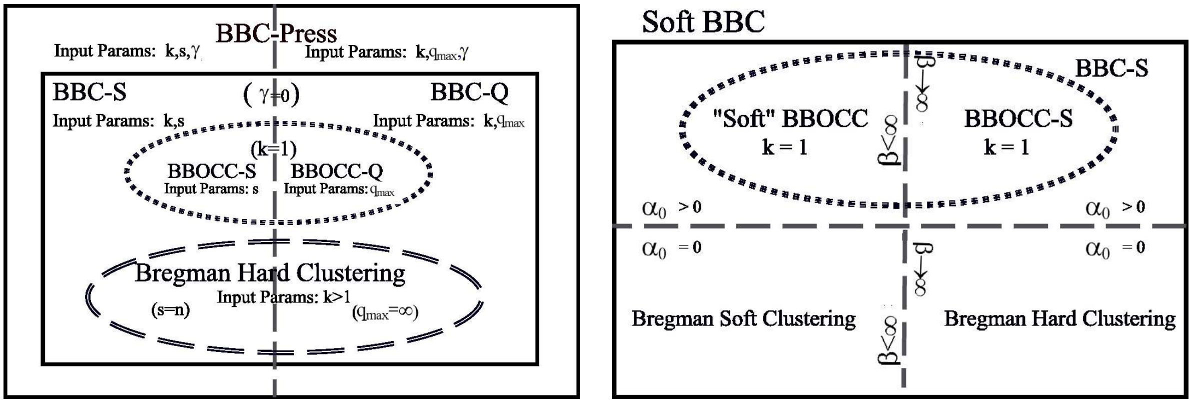


**Figure 2. Unification of various algorithms for a given Bregman divergence $D_\phi$: (left) BBC, BBOCC and Bregman Hard Clustering are special cases of BBC-Press. (right) Bregman Hard and Soft Clustering, BBC-S, BBOCC-S and a "soft" BBOCC (consisting of one exponential and a uniform background mixture) are special cases of Soft BBC obtained as specific combinations of (i) whether $\beta \to \infty$, (ii) whether $\alpha_0$ is 0 (equation 3), and (iii) whether $k$ is 1. Bregman Clustering (both hard and soft) for $k = 1$ does not result in a useful algorithm. BBOCC-S and BBOCC-Q represent BBOCC with fixed $s$ or $q_{max}$ as inputs respectively.**

whereas BBC gives rise to BBOCC. In the context of finding dense regions in the data, BBC can be thought of as a conceptual bridge between the problem of one class clustering and exhaustive k class clustering. However, the defining characteristic of BBC is its ability to find small, dense regions by modeling a small subset of the data. BBC combines the salient characteristics of both Bregman Hard Clustering and BBOCC resulting in an algorithm more powerful than either, and that works across all Bregman divergences. BBC-S is a natural extension of BBOCC-S following directly from a common underlying generative model, and is not just a heuristic; the difference in the generative model is only in having a single vs. multiple exponential distributions mixed with a uniform background.

# 9 Experiments

## 9.1 Datasets

**Table 1. A summary of the datasets used. Mic. stands for gene-expression data from microarray experiments, Sim. for artificial/simulated data, Sq. E. stands for Squared Euclidean, and $D$ is the distance function used for clustering. $|\mathcal{C}|$ is the number of classes in the data.**

| Dataset | Source | $n$ | $d$ | $D$ | $\|\mathcal{C}\|$ |
|---|---|---|---|---|---|
| Gasch Array | Mic. | 173 | $6,151$ | $D_P$ | 12 |
| Lee | Mic. | 5,612 | 591 | $D_P$ | NA |
| Gauss-2 | Sim. | 1,298 | 2 | Sq. E. | 5 |
| Gauss-10 | Sim. | 2,600 | 10 | Sq. E. | 5 |
| Gauss-40 | Sim. | 1,298 | 40 | Sq. E. | 5 |

Table 1 describes the essential attributes of the datasets that we report results on. The Gauss-2 dataset was generated using five 2-D Gaussians of different variances (Figure 3) and a uniform distribution. Similar datasets were generated with five Gaussians in 10-D and 40-D to produce Gauss-10 and Gauss-40 datasets. These datasets are useful for verifying algorithms since the true labels are known exactly. Both Gasch Array [6] and Lee [13] are yeast microarray datasets. The Gasch Array dataset contains labels for experiments, and is therefore useful for evaluating clustering of experiments in a very high-dimensional (6,151) space. The Lee dataset consists of 591 gene-expression experiments on 5,612 yeast genes obtained from the Stanford Microarray database [7] (http://genome-www5.stanford.edu/) and also contains a *Gold* standard based on Gene Ontology (GO) annotations (http://www.geneontology.org). The Gold standard contains 121,406 pairwise links (out of a total of 15,744,466 gene pairs) between 5,612 genes in the Lee data that are known to be functionally related.

## 9.2 Evaluation Methodology

**Evaluation Criteria**: Evaluating clustering is a challenging problem since the clustering itself is unsupervised and there is no direct way of identifying correspondence between class labels and clusters. Besides using the internal cost measure $Q$, we also performed three different types of evaluations based upon the type of labeled data: (1) *Adjusted Rand Index* (ARI) [10], which returns 1 for a perfect agreement between clusters and class labels and 0 when the clustering is as bad as random assignments. (2) *p-value*: We obtained p-values for individual clusters of Yeast genes using *Funspec* (http://funspec.med.utoronto.ca/) [16].

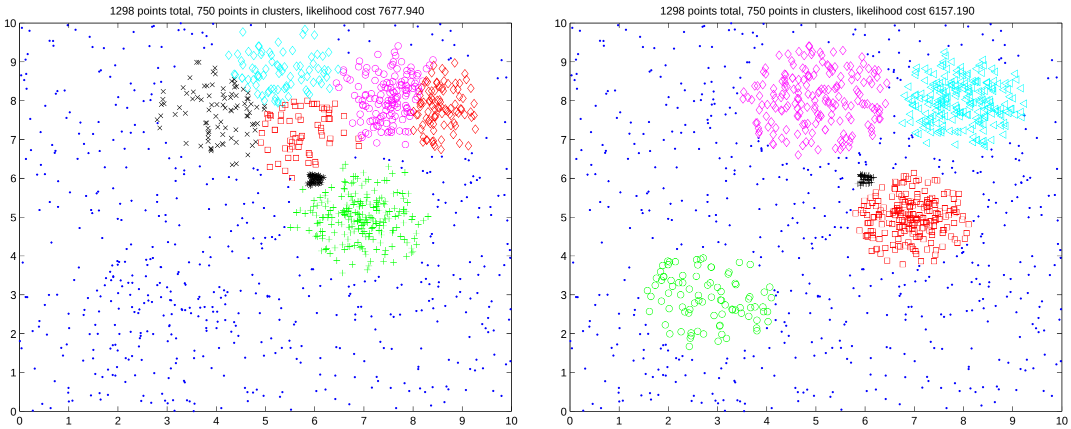


**Figure 3. Illustration of Gaussian Bubbles generated by Soft BBC-Press on Gauss-2 data when variances are updated for $s$=750, for $k = 7$ (left), and $k = 5$ (right). $k$ was set to 7 in the left figure to illustrate non-linear boundaries produced in Soft BBC. Small dots are points in the "don't-care" set.**

(3) *Overlap Lift*: It is not possible to use ARI to evaluate against the links in the Lee Gold standard. Instead, we compute statistical significance as follows: $k$ clusters of size $\{w_j\}_{j=1}^k$ result in $l_c = \sum_{j=1}^k w_j(w_j - 1)/2$ links. If $l_{true}$ is the number of correct links observed then Overlap Lift $= l_{true}/(f_{linked} l_c)$, representing how many times more correct links are observed as compared to random chance, where $f_{linked}$ is the fraction of gene pairs linked in the Gold standard.

For all evaluations, the points in the background or the "don't care" cluster are excluded from the evaluation. Note that the clustering is performed in a completely unsupervised setting and the class labels were only used for evaluation.

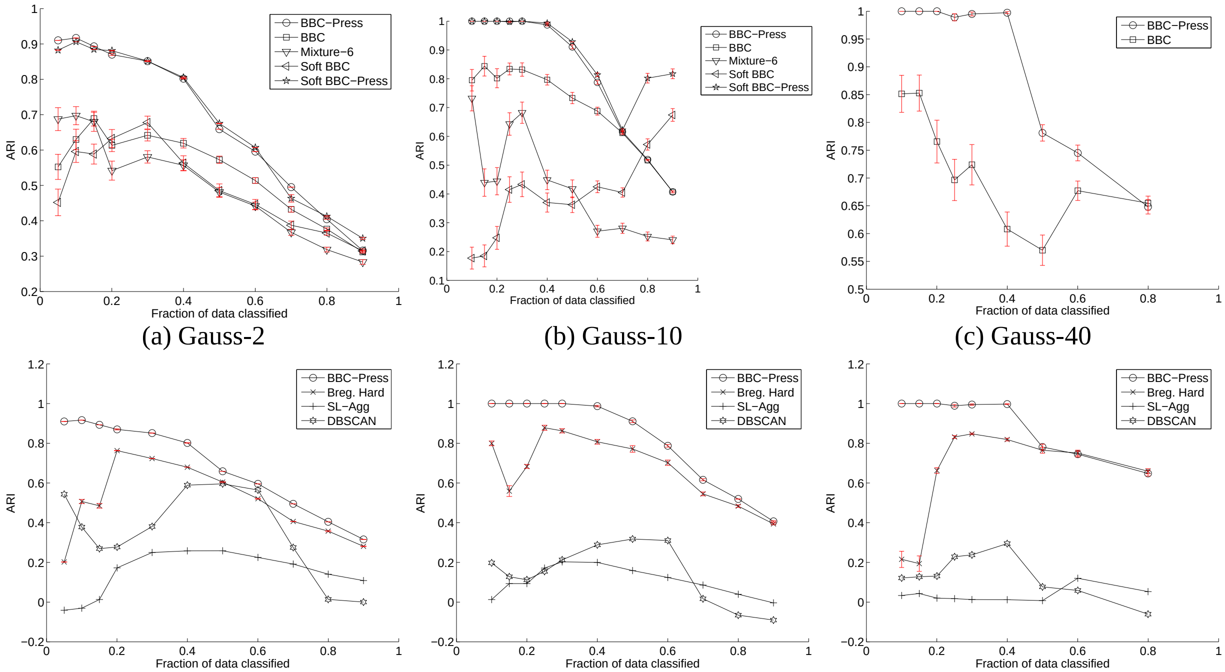
**Evaluating Soft BBC**: We tested Soft BBC with Gaussians as the exponential mixture components. If the Gaussian variances are treated as a part of the mixture parameters $\{\theta_j\}_{j=1}^k$ (equation 3), it is possible to get clusters of variable diameters (Figure 3, right) that fit natural cluster diameters. There are eight possible variations of Soft BBC depending upon whether (i) $\alpha_0$ is updated (Case A vs. B, Section 5.4), (ii) variances are updated or not, and (iii) all cluster variances are forced to be equal or not. We present results on the Soft BBC implementation that gives the best results: updatable, unequal variances with a fixed $\alpha_0$. We also compared Soft BBC for Gaussians with an alternative model that we call *Mixture-6* where the uniform background distribution is replaced by a large, fixed variance Gaussian while the other $k$ Gaussian variances are updated, and $\alpha_0$ is fixed.

**Hard Assignments for Soft BBC**: On convergence, the points are assigned to the mixture with the largest probability. A post-processing was performed that recomputes $p_0$ such that exactly $(n-s)/n$ points are assigned to the "don't care" set. A similar conversion was required for evaluating the soft model Mixture-6.

**Comparison with other methods**: We also compared our method with Bregman Hard Clustering, Single Link Agglomerative clustering and DBSCAN. Bregman Hard Clustering assigns every data point into a cluster. To be able to compare it meaningfully with BBC, we picked $s$ points closest to their respective cluster representatives. This procedure was also used for Single Link Agglomerative clustering. For the two DBSCAN parameters, we set $MinPts$ to 4 as recommended by Ester et al. [5], while we searched for $Eps$ that resulted in $s$ points in clusters. $k$ is automatically estimated by DBSCAN while for all the other methods and datasets $k$ was set to $|\mathcal{C}|$ (Table 1), except for the Lee dataset (where $|\mathcal{C}|$ is not known) where we set $k$ to 10. All five methods use the same (and the appropriate) distance measure; Sq. Euclidean for the Gaussian and Pearson Distance for the gene-expression datasets respectively.

## 9.3 Results

**Pressurization with Soft BBC**: For the lower dimensional datasets, Soft BBC-Press, does extremely well, giving near-perfect results (ARI $\approx$ 1) for up to 40% coverage on Gauss-10 data and an ARI between 0.8 and 0.9 for up to 40% coverage on Gauss-2 data. We only tested the Soft BBC and the Mixture-6 models on Gauss-2 and Gauss-10 datasets, mainly to validate Soft BBC and Soft BBC-Press. This is because, exponential mixture models in general, including Bregman Soft Clustering, Mixture-6 and Soft BBC all suffer from an inherent flaw that makes them impractical for high dimensional datasets; there are rounding errors while estimating the mixture membership probabilities (equation 7), and these rounding errors worsen exponentially with the dimensionality of the data $d$, so much so that the models generally do not work well beyond $d = 10$. However, the main purpose of designing Soft BBC was



(a) Gauss-2 (b) Gauss-10 (c) Gauss-40

(d) Gauss-2 (e) Gauss-10 (f) Gauss-40

**Figure 4. Labeled evaluation on simulated Gaussian data of increasing dimensionality using ARI: (a), (b) and (c) demonstrate the effectiveness of Pressurization. (d), (e) and (f) show effectiveness of BBC-Press as compared to three other methods: Bregman Hard Clustering, Single Link Agglomerative and DBSCAN. Error bars of one std. deviation are shown (but are sometimes too small to be visible) for local search methods for which ARI is plotted as the average over 100 trials with random initialization.**

to show that a fundamental generative model lies behind BBC (Section 8). Furthermore, figure 4(a) and (b) show that Mixture-6 has no clear performance advantage over Soft BBC. Also, Mixture-6 does not conform to the form required to incorporate Pressurization and does not correspond to any known "hard" model for Bregman divergences, and a hard model is essential to scale to higher dimensional datasets.

**Pressurization with (Hard) BBC**: As predicted, both BBC and Soft BBC without Pressurization tend to be a lot more sensitive to initialization, and BBC-Press performs almost as well as Soft BBC-Press on the Gauss-2 and Gauss-10 datasets giving ARI $\approx$ 1 for coverages of up to 40%. On the Gauss-40 dataset, BBC-Press continues to give an ARI $\approx$ 1 for up to 40% coverage. In contrast, we were unable to run Soft BBC-Press for Gauss-40 dataset because of severe rounding errors. These results are impressive given that the ARI was obtained as averages of multiple runs with random seeding. In Figure 5(f), for lower coverages, BBC-Press gives significantly lower cost ($APD$) as compared to both BBC and Bregman Hard, which also used a similar cost function. The improvement against labeled data using BBC Press as compared to BBC is also dramatic for both Gasch Array and Lee, showing that Pressurization also works well for clustering high dimensional gene experiments (Figure 5(a)) or genes (5 (d)). Note that the error bars were plotted on all the local search algorithms, but are often too small to be visible.

**Comparison with other types of Algorithms**: On the Gaussian datasets (Figure 4(d) to (f)), and on the two gene-expression datasets (Figure 5(b) and (e)), DBSCAN, Single Link Agglomerative and Bregman Hard Clustering all perform much worse than BBC-Press in general, and especially when clustering a part of the data. These results are based on evaluation using labels not used for clustering; using ARI on Gaussians (Figure 4(d) to (f)) and Gasch Array (Figure 5(b)) and using Overlap Lift on Lee (Figure 5(e)), and are therefore independent of the clustering methodology. Figure 5(e) shows that (1) BBC-Press not only beats other methods by a wide margin but also shows high enrichments of links for low coverages (over 6 times for 5 % coverage), and (2) Single Link Agglomerative clustering does not work well for clustering genes and gives results not much better than random. On all datasets, Single Link tends to perform the worst; one explanation might be its inability to handle noisy data. In fact, for some situations (Figure 4(d) to (f)), DBSCAN and Single Link Agglomerative give slightly worse than random performance resulting in ARI values that are slightly below 0. The performance difference between our method (BBC-Press) and the other

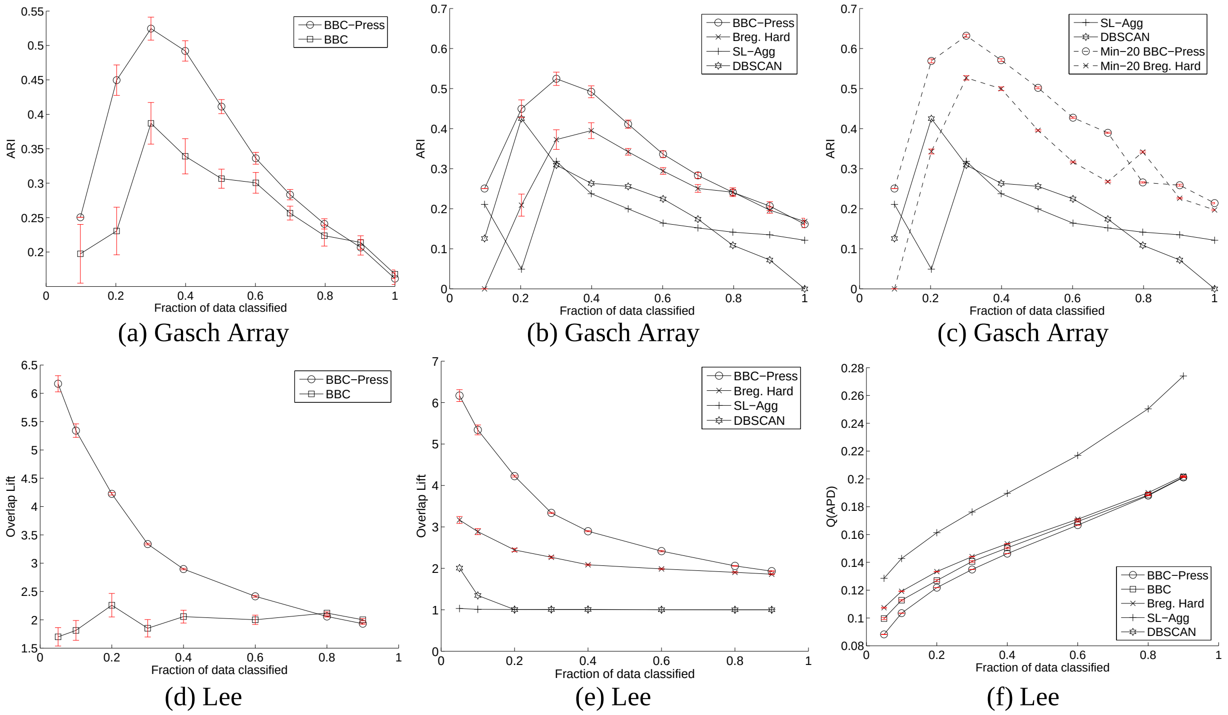


**Figure 5. Evaluation of BBC-Press on gene-expression data using ARI for Gasch Array, and Overlap Lift and internal cost (APD) for Lee, as compared to BBC, Bregman Hard Clustering, Single Link Agglomerative, and DBSCAN. Local search results were averaged over 20 trials and the corresponding one std. dev. error-bars are plotted that are sometimes too small to be visible. For (c), "Min 20" models were obtained by picking the solution with the lowest cost (APD) in 20 random seeded trials, and then picking the mean of repeating the "Min 20" model also 20 times.**

three methods is quite significant on all the five datasets, given the small error bars. Additionally, if we were to pick the minimum-cost solution out of multiple trials for the local search methods, the differences in the performance between BBC-Press vs. DBSCAN and Single Link becomes even more substantial, e.g. Figure 5, (b) vs. (c) for Gasch Array.

**Selecting size and number of dense clusters**: In BBC-Press, $s$ controls the number of data points in dense clusters. The dense clusters were invariably very pure when using BBC-Press, with near-perfect clusters on the Gaussian data for $s$ of up to 40% of $n$, while on the Gasch Array dataset the performance peaks at a coverage of around 0.3 but shows a general decline after that. The rapid increase in cluster quality with decreasing $s$ is more pronounced in BBC-Press than in the other methods, and shows that on these datasets, dense regions are indeed highly correlated with the class labels; the confirmation of this phenomena is tantalizing considering the fact that the clustering process was completely unsupervised. In practice, selecting dense clusters with BBC-Press requires choosing an appropriate $s$ and $k$. If small amounts of labeled data is available, the best $k$ can be estimated for a fixed $s$ using an approach such as PAC-MDL [1], while a reasonable $s$ can be picked by applying BBC-Press on a range of $s$ and picking the "knee" (e.g. Figures 4(a),(b),(c) and 5(c) show a sudden decline in ARI near $s = 0.4 \times n$). Alternatively, in many problems $k$ can be an input, while $s$ simply has to be a small threshold (e.g. for finding a small number of relevant web documents, or a small number of relevant genes (Figure 5(e)).

**Visual Verification**: Although the results based on performance measures show the effectiveness of our method, visual verification serves as another independent validation that the clusters are not only statistically significant but also useful in practice. For the Gauss-2 dataset, it is easy to verify the quality of the clusters visually (Figure 3). For the Gasch Array clustering, most clusters were generally very pure using BBC-Press for lower coverages. For example, when only 70 out of 173 experiments are clustered by repeating BBC-Press 20 times and picking the lowest cost solution, the average ARI is around 0.6 over 12 classes. Some clusters are even purer, for example, one of the clusters contained 12 out of 13 points belonging to the class "YPD", while there are 22 experiments of type YPD. This gives us an accuracy of 92.31% for a coverage of 0.591 when 40 % of the data was clustered into 12 clusters. Similarly, for the Lee dataset, we verified a high purity cluster using FunSpec; 10 out of 14 genes in one of the clusters belonged to the functional category “cytoplasmic and nuclear degradation” with a p-value of $< 10^{-14}$, the probability of this cluster belonging into to the category by random chance. Many other gene clusters on the Lee dataset also had low p-values for some of the categories recovered by FunSpec.

## 10 Concluding Remarks

Empirical results show that BBC-Press outperforms other potential alternatives by a large margin and gives good results on a variety of problems involving low to very high-dimensional feature spaces. BBC-Press can be seen as a powerful extension of One Class Clustering to a multi-class setting where the goal is to find dense regions in the data. Our method extends the notion of “density-based clustering” to a large class of divergence measures, and is perhaps the first that uses a local search/parametric approach. The low time and space complexity of the local search approach, coupled with the robustness provided by Pressurization, makes it possible to find multiple dense regions on extremely large and high-dimensional datasets, thus opening density-based clustering to much larger problems. Bregman Bubble Clustering can also be thought of as a conceptual bridge between partitional clustering algorithms and the problem of One Class Clustering. The Soft BBC model shows that BBC arises out of a more fundamental model involving a mixture of exponentials and a uniform background, and explains why BBC performs better than Bregman Clustering by incorporating a model for the “noisy” background. The extension of BBC to Pearson Correlation (Pearson Distance) makes it applicable to a variety of biological datasets where finding small, dense clusters is critical.

**Acknowledgments**: This research was supported by NSF grants IIS-0325116 and IIS-0307792. We are also grateful to Srujana Merugu for some useful discussions.

## References

[1] A. Banerjee and J. Langford. An objective evaluation criterion for clustering. In *KDD-04*, Seattle, Washington, USA, August 2004.

[2] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *JMLR*, 6:1705–1749, 2005.

[3] K. Crammer and G. Chechik. A needle in a haystack: Local one-class optimization. In *In Proc. ICML*, Banff, Alberta, Canada, 2004.

[4] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2):143–175, January-February 2001.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *In Proc. KDD-96*, 1996.

[6] Gasch A. P. et al. Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Bio. of the Cell*, 11(3):4241–4257, December 2000.

[7] Gollub J. et al. The stanford microarray database: data access and quality assessment tools. *Nucleic Acids Res*, 31:94–96, 2003.

[8] G. Gupta and J. Ghosh. Robust one-class clustering using hybrid global and local search. In *Proc. ICML 2005*, pages 273–280, Bonn, Germany, August 2005.

[9] Hastie T. et al. Gene shaving as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1:1–21, 2000.

[10] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, pages 193–218, 1985.

[11] D. Jiang, J. Pei, and A. Zhang. DHC: A density-based hierarchical clustering method for time series gene expression data. In *BIBE ’03*, page 393, Washington, DC, USA, 2003. IEEE Comp. Soc.

[12] M. Kearns, Y. Mansour, and A. Y. Ng. An information-theoretic analysis of hard and soft assignment methods for clustering. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 282–293. AAAI, 1997.

[13] I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306:1555–1558, 2004.

[14] R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*. Kluwer, 1998.

[15] S. D. Pietra, V. D. Pietra, and J. Lafferty. Duality and auxiliary functions for Bregman distances. In *Technical Report CMU-CS-01-109, School of Computer Science*, Carnegie Mellon University, 2001.

[16] M. D. Robinson, J. Grigull, N. Mohammad, and T. R. Hughes. Funspec: a web-based cluster interpreter for yeast. *BMC Bioinformatics*, 35(3), November 13 2002.

[17] B. Schölkopf, J. C. Platt, J. S. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[18] D. Tax and R. Duin. Data domain description using support vectors. In *Proceedings of the ESANN-99*, pages 251–256, 1999.