Scaling clustering algorithms with Bregman distances[1]

Jacob Kogan
kogan@umbc.edu
Department of Mathematics and Statistics
UMBC, Baltimore, MD 21250
and
Marc Teboulle
teboulle@post.tau.ac.il
School of Mathematical Sciences
Tel-Aviv University
Tel-Aviv, Israel

Abstract

Clustering algorithms often require to keep the entire dataset in the computer memory. When the dataset is large and does not fit into available memory one has to "squash" the dataset to make applications of $k-$means like algorithms possible. The Balanced Iterative Reducing and Clustering algorithm (BIRCH) is a clustering algorithm designed to operate under the assumption "the amount of memory available is limited, whereas the dataset can be arbitrary large" [22]. The algorithm does the "squashing", or generates "a compact dataset summary" minimizing I/O cost involved. An application of quadratic batch $k-$means to the BIRCH generated "summary" is proposed in [4]. The present note combines BIRCH and $k-$means clustering equipped with Bregman distances. We report preliminary numerical experiments on two small datasets, so that the results of clustering with and without BIRCH can be compared. The suggested BIRCH + $k-$means clustering scheme combines batch and incremental iterations, allows a choice of a variety of distance like functions, and may be useful for clustering large data collections, like, for example, the Enron dataset.

1. Introduction

The concept of divergence measures, also called distance like functions, which are derived from given convex functions, have been introduced by Bregman [5] and Csiszar [11]. These distance like functions have been extended and successfully used in the context of optimization theory and algorithms in many studies (see e.g. [6], [7] for Bregman distances and [17], [16], [18] for Csiszar based divergences, and references therein). In a number of recent publications divergence measures have also been applied and shown to be useful in machine learning problems (for instance, [15], [14], [9], [10], [21]) and in clustering (for example, [12], [13], [1]).

In this paper we extend the Bradley–Fayyad–Reina [4] idea of applying the classical quadratic batch $k-$means algorithm to clusters (rather than vectors) generated by BIRCH, by replacing the quadratic Euclidean distance with a Bregman divergence. In addition we augment batch $k-$means by incremental iterations (see [19]), thus improving final partitions generated by the algorithm. Numerical experiments reported in the paper are performed on a sparse data typical for IR applications. We analyze sparsity of partitions generated by BIRCH, the number of iterations performed and the quality of partitions generated by $k-$means with and without BIRCH.

[1] Research supported in part by the United States–Israel Binational Science Foundation (BSF), Grant No. 2010

2. Setting

For a set of vectors $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_m\} \subset \mathbf{R}^n$, a prescribed subset $\mathcal{C}$ of $\mathbf{R}^n$ and a "distance" function $d(\mathbf{x}, \mathbf{a})$ define a centroid $\mathbf{c} = \mathbf{c}(\mathcal{A})$ of the set $\mathcal{A}$ as a solution of the minimization problem

$$\mathbf{c} = \arg\min \left\{ \sum_{\mathbf{a}\in\mathcal{A}} d(\mathbf{x}, \mathbf{a}),\ \mathbf{x} \in \mathcal{C} \right\}. \tag{2.1}$$

The quality of the set $\mathcal{A}$ is denoted by $Q(\mathcal{A})$ and is defined by

$$Q(\mathcal{A}) = \sum_{i=1}^{m} d(\mathbf{c}, \mathbf{a}), \quad \text{where } \mathbf{c} = \mathbf{c}(\mathcal{A}) \tag{2.2}$$

(we set $Q(\emptyset) = 0$ for convenience). Let $\Pi = \{\pi_1, \ldots, \pi_k\}$ be a partition of $\mathcal{A}$, i.e.

$$\bigcup_i \pi_i = \mathcal{A}, \text{ and } \pi_i \cap \pi_j = \emptyset \text{ if } i \neq j.$$

We abuse notations and define the quality of the partition Π by

$$Q(\Pi) = Q(\pi_1) + \ldots + Q(\pi_k). \tag{2.3}$$

We aim to find a partition $\Pi^{\min} = \{\pi_1^{\min}, \ldots, \pi_k^{\min}\}$ that *minimizes* the value of the objective function Q. The problem is known to be NP–hard, and we are looking for algorithms that generate "reasonable" solutions.

It is easy to see that centroids and partitions are associated as follows:

1. Given a partition $\Pi = \{\pi_1, \ldots, \pi_k\}$ of the set $\mathcal{A}$ one can define the corresponding centroids $\{\mathbf{c}(\pi_1), \ldots, \mathbf{c}(\pi_k)\}$ by:

$$\mathbf{c}(\pi_i) = \arg\min \left\{ \sum_{\mathbf{a}\in\pi_i} d(\mathbf{x}, \mathbf{a}),\ \mathbf{x} \in \mathcal{C} \right\}. \tag{2.4}$$

2. For a set of k "centroids" $\{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$ one can define a partition $\Pi = \{\pi_1, \ldots, \pi_k\}$ of the set $\mathcal{A}$ by:

$$\pi_i = \left\{ \begin{array}{c} \mathbf{a} : \mathbf{a} \in \mathcal{A},\ \ d(\mathbf{c}_i, \mathbf{a}) \leq d(\mathbf{c}_l, \mathbf{a}) \\ \text{for each } l = 1, \ldots, k \end{array} \right\} \tag{2.5}$$

(we break ties arbitrarily). Note that, in general, $\mathbf{c}(\pi_i) \neq \mathbf{c}_i$.

The batch k–means algorithm is a procedure that iterates between the two steps described above to generate a partition Π' from a partition Π.

Note that when the dataset $\mathcal{A}$ does not fit into available computer memory evaluation of (2.5) becomes problematic. A possible solution to this problem for the particular choice $d(\mathbf{x}, \mathbf{a}) = \|\mathbf{x} - \mathbf{a}\|^2$ is suggested in [22] as follows: Let $\Pi = \{\pi_1, \ldots, \pi_M\}$ be a partition of $\mathcal{A}$. If for $i = 1, \ldots, M$

1. $\mathbf{b}_i = \mathbf{c}(\pi_i) = \frac{1}{m_i} \sum_{\mathbf{a}\in\pi_i} \mathbf{a}$ the centroid of π_i,
2. $m_i = m(\mathbf{b}_i) = |\pi_i|$ the size of π_i,
3. $q_i = Q(\pi_i)$ the quality of π_i,

then

$$Q(\pi_{i_1} \cup \ldots \cup \pi_{i_p}) = \sum_{j=1}^{p} q_{i_j} + \sum_{j=1}^{p} m_{i_j} \|\mathbf{c} - \mathbf{b}_{i_j}\|^2, \tag{2.6}$$

where

$$\mathbf{c} = \frac{m_{i_1}\mathbf{b}_{i_1} + \ldots + m_{i_p}\mathbf{b}_{i_p}}{m_{i_1} + \ldots + m_{i_p}}.$$

Formula (2.6) paves the way to approach the clustering of $\mathcal{A}$ along the two lines:

1. Given a positive real constant R (that controls the "spread" of a cluster), an integer L (that controls the size of a cluster), p already available clusters $\pi_1, \ldots \pi_p$ (i.e. p triplets

$(m_i, q_i, \mathbf{b}_i))$, and a vector $\mathbf{a} \in \mathbf{R}^n$ one can compute $Q(\pi_i \cup \{\mathbf{a}\})$, $i = 1, \ldots, p$. If for some index i

$$Q(\pi_i \cup \{\mathbf{a}\}) < R \text{ and } m_i + 1 \leq L,$$

then $\mathbf{a}$ is assigned to π_i, and the triplet $(m_i, q_i, \mathbf{b}_i)$ is updated. Otherwise $\{\mathbf{a}\}$ becomes a new cluster π_{p+1} (this is the basic BIRCH construction).

2. Once a partition $\Pi = \{\pi_1, \ldots, \pi_M\}$ of $\mathcal{A}$ is available one can cluster the set $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_M\}$. Note that the M cluster partition $\{\pi_1, \ldots, \pi_M\}$ of $\mathcal{A}$ associates each subset $\pi^{\mathcal{B}} \subseteq \mathcal{B}$ with a subset $\pi^{\mathcal{A}} \subseteq \mathcal{A}$ through

$$\pi^{\mathcal{A}} = \bigcup_{\mathbf{b}_j \in \pi^{\mathcal{B}}} \pi_j.$$

   Hence a k cluster partition $\Pi_{\mathcal{B}} = \{\pi_1^{\mathcal{B}}, \ldots, \pi_k^{\mathcal{B}}\}$ of the set $\mathcal{B}$ can be associated with a k cluster partition $\Pi_{\mathcal{A}} = \{\pi_1^{\mathcal{A}}, \ldots, \pi_k^{\mathcal{A}}\}$ of the set $\mathcal{A}$ through

$$\pi_i^{\mathcal{A}} = \bigcup_{\mathbf{b}_j \in \pi_i^{\mathcal{B}}} \pi_j, \; i = 1, \ldots, k. \qquad (2.7)$$

   One can, therefore, apply quadratic $k-$means to a smaller dataset $\mathcal{B}$ to generate a partition of the dataset $\mathcal{A}$ (this approach is suggested in [4]).

In this paper we exploit a key result reported in [1] along with the tools developed in [19] and extend the above approaches to clustering with Bregman distances.

3. Bregman distance

We start with a brief introduction of Bregman divergences (for detailed account and discussion see e.g. [19]). Let $\psi : \mathbf{R}^n \to \mathbf{R}^n \cup \{+\infty\}$ be a proper strictly convex function which is assumed continuously differentiable on the interior of its effective domain int dom $\psi = S$, assumed to be nonempty. The Bregman distance (also called "Bregman divergence") $D_\psi : \text{cl}S \times S \to [0, +\infty)$ is defined by

$$D_\psi(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{x}) - \psi(\mathbf{y}) - \langle \nabla\psi(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle, \quad (3.1)$$

where $\nabla\psi$ is the gradient of ψ, and $\langle \cdot, \cdot \rangle$ stands for the inner product in $\mathbf{R}^n$. This function measures the convexity of ψ, i.e. $D_\psi(\mathbf{x}, \mathbf{y}) \geq 0$ if and only if the gradient inequality for ψ holds, i.e., if and only if ψ is convex, see Figure 1.

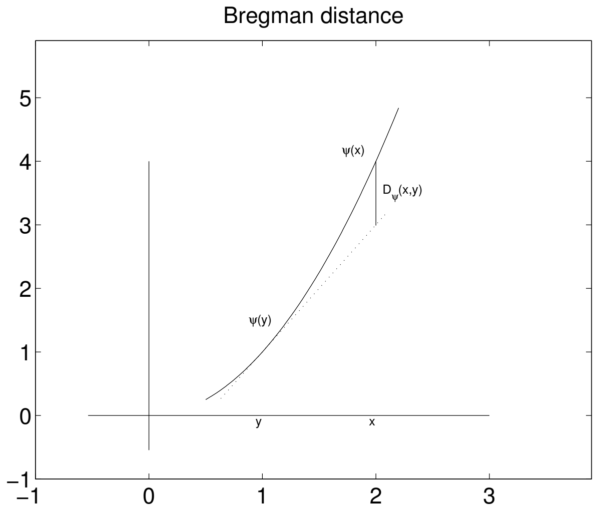

Figure 1: Bregman divergence

Two examples of well known Bregman based distance like functions are given in Table 1. Distance like functions are not necessarily symmetric (hence the "distance like" attribute). This lack of symmetry allows for considering Bregman distances with a change in the order of the variables in D_ψ, i.e.,

$$\overleftarrow{D_\psi}(\mathbf{x}, \mathbf{y}) = \psi(\mathbf{y}) - \psi(\mathbf{x}) - \langle \nabla\psi(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \quad (3.2)$$

$\psi(\mathbf{x})$	$\lVert\mathbf{x}\rVert^2$	$\sum_{j=1}^{n} \mathbf{x}[j]\log\mathbf{x}[j]-\mathbf{x}[j]$
$D_\psi(\mathbf{x},\mathbf{y})$	$\lVert\mathbf{x}-\mathbf{y}\rVert^2$	$\sum_{j=1}^{n} \mathbf{x}[j]\log\frac{\mathbf{x}[j]}{\mathbf{y}[j]}$ $+\mathbf{y}[j]-\mathbf{x}[j]$

Table 1: kernels and divergences

(compare with (3.1)). Then, for example, by using the kernel

$$\psi(\mathbf{x})=\frac{\nu}{2}\lVert\mathbf{x}\rVert^2+\mu\left[\sum_{j=1}^{n}\mathbf{x}[j]\log\mathbf{x}[j]-\mathbf{x}[j]\right], \tag{3.3}$$

with $\nu \geq 0$, $\mu \geq 0$ we obtain

$$\begin{aligned}\overleftarrow{D}_\psi(\mathbf{x},\mathbf{y}) = D_\psi(\mathbf{y},\mathbf{x}) &= \frac{\nu}{2}\lVert\mathbf{y}-\mathbf{x}\rVert^2\\ + \quad \mu\sum_{j=1}^{n}&\left[\mathbf{y}[j]\log\frac{\mathbf{y}[j]}{\mathbf{x}[j]}+\mathbf{x}[j]-\mathbf{y}[j]\right] \end{aligned}\tag{3.4}$$

(numerical experiments reported in this paper are conducted with this distance like function with various values for ν and μ).

Note that by changing the order of the variables, the convexity of $\mathbf{x} \to \overleftarrow{D}_\psi(\mathbf{x},\mathbf{y})$ is not anymore necessarily warranted.[2] Despite this lack of convexity in general, a surprising and key result for centroids computation is reported in [1]:

Theorem 3.1 *(Banerjee et al.)*
If $\mathbf{z} = \dfrac{\mathbf{a}_1+\ldots+\mathbf{a}_m}{m}$, *then* $\sum_{i=1}^{m} D_\psi(\mathbf{a}_i,\mathbf{z}) \leq \sum_{i=1}^{m} D_\psi(\mathbf{a}_i,\mathbf{x})$.

The result shows that the solution to the generally nonconvex optimization problem (2.4) with the distance like function $d(\mathbf{x},\mathbf{a}) = \overleftarrow{D}_\psi(\mathbf{x},\mathbf{a}) = D_\psi(\mathbf{a},\mathbf{x})$ is always given by the arithmetic mean. This result paves the way to the development of $k-$means clustering with such Bregman distances. A generalization of (2.6) useful for extending BIRCH type clustering to datasets equipped with Bregman distances is reported in [19]:

Theorem 3.2 *(Teboulle et al.)*
If $\mathcal{A} = \pi_1 \cup \pi_2 \cup \ldots \cup \pi_k$ *with* $m_i = |\pi_i|$, $\mathbf{c}_i = \mathbf{c}(\pi_i)$, $i = 1,\ldots,k$;

$$\mathbf{c} = \mathbf{c}(\mathcal{A}) = \frac{m_1}{m}\mathbf{c}_1 + \ldots + \frac{m_k}{m}\mathbf{c}_k,$$

where $m = m_1+\ldots+m_k$, *and* $\Pi = \{\pi_1\,\pi_2,\ldots,\pi_k\}$, *then*

$$\begin{aligned} Q(\Pi) &= \sum_{i=1}^{k} Q(\pi_i) + \sum_{i=1}^{k} m_i d(\mathbf{c},\mathbf{c}_i)\\ &= \sum_{i=1}^{k} Q(\pi_i) + m_i\left[\psi(\mathbf{c}_i)-\psi(\mathbf{c})\right]. \end{aligned}\tag{3.5}$$

4. Clustering procedure

The proposed clustering procedure consists of the following steps:

1. Apply BIRCH type procedure to the dataset $\mathcal{A} = \{\mathbf{a}_1,\ldots,\mathbf{a}_m\} \subset \mathbf{R}^n$ to generate a vector set $\mathcal{B} = \{\mathbf{b}_1,\ldots,\mathbf{b}_M\} \subset \mathbf{R}^n$, $M < m$ along with scalars q_i, m_i, $i = 1,\ldots,M$.

2. Apply PDDP (see [3]) to $\mathcal{B}$ to generate the initial partition $\Pi_{\mathcal{B}} = \{\pi_1^{\mathcal{B}},\ldots,\pi_k^{\mathcal{B}}\}$ of the set $\mathcal{B}$ to be used later by $k-$means clustering.

3. Apply batch $k-$means followed by incremental $k-$means (see [19] for details) to the initial partition $\Pi_{\mathcal{B}} = \{\pi_1^{\mathcal{B}},\ldots,\pi_k^{\mathcal{B}}\}$ (in the sequel we refer to this procedure simply as $k-$means).

[2] However, note that for the proposed example (3.4) the convexity in the $\mathbf{x}$ argument is preserved. In fact, for this example, D_ψ is *jointly* convex in both variables, see [16].

In what follows we provide a brief description of the first and the last steps of the procedure (application of PDDP to the set $\mathcal{B}$ is straightforward).

4.1. BIRCH type procedure

Given a real constant $R > 0$ and an integer $L > 0$ we would like to build a partition Π of $\mathcal{A}$ so that

$$\Pi = \{\pi_1, \ldots, \pi_M\}, \; Q(\pi_i) < R, \; |\pi_i| \le L \qquad (4.1)$$

for each $i = 1, \ldots, M$. The procedure starts by picking an arbitrary vector $\mathbf{a} \in \mathcal{A}$ and building a one cluster singleton partition $\Pi = \{\pi_1\}$ with $\pi_1 = \{\mathbf{a}\}$. If a partition $\Pi = \{\pi_1, \ldots, \pi_p\}$ is already available and there is a vector $\mathbf{a} \in \mathcal{A}$ that does not belong to $\bigcup\limits_{i=1}^{p} \pi_i$, then the partition Π is updated:

1. Identify an index $1 \le i \le p$ so that

   $$Q(\pi_i \cup \{\mathbf{a}\}) < R \text{ and } |\pi_i| + 1 \le L. \qquad (4.2)$$

   Assign $\mathbf{a}$ to π_i and update $\mathbf{b}_i$, q_i and m_i.

2. If (4.2) fails for each $1 \le i \le p$ a new singleton cluster $\pi_{p+1} = \{\mathbf{a}\}$ is formed, and the updated partition becomes $\Pi = \{\pi_1, \ldots, \pi_p, \pi_{p+1}\}$.

Due to Theorem 3.2 one has

$$Q(\pi_i \cup \{\mathbf{a}\}) = q_i + m_i d(\mathbf{c}, \mathbf{b}_i) + d(\mathbf{c}, \mathbf{a}), \qquad (4.3)$$

where

$$\mathbf{c} = \frac{m_i}{m_i + 1}\mathbf{b}_i + \frac{1}{m_i + 1}\mathbf{a}. \qquad (4.4)$$

Hence, to check (4.2) one needs an access to $\mathbf{b}_i$, q_i, m_i, and $\mathbf{a}$ (the vectors contained in π_i are not needed). Furthermore (4.3) provides an update for q_i, and (4.4) updates $\mathbf{b}_i$.

4.2. $k-$means

Consider a k cluster partition $\Pi_{\mathcal{B}} = \{\pi_1^{\mathcal{B}}, \ldots, \pi_k^{\mathcal{B}}\}$ of the set $\mathcal{B}$ and the induced k cluster partition $\Pi_{\mathcal{A}} = \{\pi_1^{\mathcal{A}}, \ldots, \pi_k^{\mathcal{A}}\}$ of the set $\mathcal{A}$ (see (2.7)). Consider $\pi_1^{\mathcal{B}}$ with $\mathbf{c}\left(\pi_1^{\mathcal{B}}\right)$ and the corresponding cluster $\pi_1^{\mathcal{A}} = \{\pi_1, \ldots, \pi_p\}$. In what follows we shall denote the number of vectors in cluster $\pi \in \Pi$ with centroid $\mathbf{b}$ by $m(\mathbf{b})$. Due to (3.5)

$$Q\left(\pi_1^{\mathcal{A}}\right) = \sum_{j=1}^{p} Q(\pi_j) + \sum_{\mathbf{b} \in \pi_1^{\mathcal{B}}} m(\mathbf{b}) d\left(\mathbf{c}\left(\pi_1^{\mathcal{B}}\right), \mathbf{b}\right).$$

Repetition of this argument for other clusters $\pi_i^{\mathcal{B}}$ and summing the corresponding expressions leads to

$$\begin{aligned} \sum_{i=1}^{k} Q\left(\pi_i^{\mathcal{A}}\right) &= \sum_{l=1}^{M} Q(\pi_l) \\ &+ \sum_{i=1}^{k} \sum_{\mathbf{b} \in \pi_i^{\mathcal{B}}} m(\mathbf{b}) d\left(\mathbf{c}\left(\pi_i^{\mathcal{B}}\right), \mathbf{b}\right). \end{aligned}$$

We set $Q_{\mathcal{B}}\left(\Pi_{\mathcal{B}}\right) = \sum\limits_{i=1}^{k} \sum\limits_{\mathbf{b} \in \pi_i^{\mathcal{B}}} m(\mathbf{b}) d\left(\mathbf{c}\left(\pi_i^{\mathcal{B}}\right), \mathbf{b}\right)$, note that $\sum\limits_{l=1}^{M} Q(\pi_l) = Q(\Pi)$ is a constant, and arrive at the following formula

$$Q\left(\Pi_{\mathcal{A}}\right) = Q(\Pi) + Q_{\mathcal{B}}\left(\Pi_{\mathcal{B}}\right). \qquad (4.5)$$

If $\Pi_{\mathcal{B}}^{(t)}$ is a sequence of partitions generated from $\Pi_{\mathcal{B}}$ by iterations of the batch $k-$means, then

$$Q_{\mathcal{B}}\left(\Pi_{\mathcal{B}}^{(t)}\right) \ge Q_{\mathcal{B}}\left(\Pi_{\mathcal{B}}^{(t+1)}\right),$$

and, due to (4.5),

$$Q\left(\Pi_{\mathcal{A}}^{(t)}\right) \ge Q\left(\Pi_{\mathcal{A}}^{(t+1)}\right).$$

Hence application of batch $k-$means to the smaller set $\mathcal{B}$ generates a sequence of better and better quality partitions of the larger set $\mathcal{A}$.

A sequence of batch $k-$means iterations augmented by an incremental iteration leads to better quality partitions sometimes without additional computational effort (see [19] for details). An incremental iteration removes a vector $\mathbf{b}$ from a cluster $\pi_i^{\mathcal{B}}$ and assigns the vector to a cluster $\pi_j^{\mathcal{B}}$. We denote $\pi_i^{\mathcal{B}}$ without $\mathbf{b}$ by $\pi_i^{\mathcal{B}-}$, $\pi_j^{\mathcal{B}}$ with the additional vector by $\pi_j^{\mathcal{B}+}$. The centroids of the four clusters are denoted by $\mathbf{c}_i$, $\mathbf{c}_i^-$, $\mathbf{c}_j$, and $\mathbf{c}_j^+$ respectively. Using (3.5) we obtain

$$\begin{aligned}&\left[Q_{\mathcal{B}}\left(\pi_i^{\mathcal{B}}\right) - Q_{\mathcal{B}}\left(\pi_i^{\mathcal{B}-}\right)\right] + \\ &\qquad\qquad \left[Q_{\mathcal{B}}\left(\pi_j^{\mathcal{B}}\right) - Q_{\mathcal{B}}\left(\pi_j^{\mathcal{B}+}\right)\right] = \\ &[M_i - m(\mathbf{b})]\left[\psi(\mathbf{c}_i^-) - \psi(\mathbf{c}_i)\right] - m(\mathbf{b})\psi(\mathbf{c}_i) + \\ &\quad [M_j + m(\mathbf{b})]\left[\psi(\mathbf{c}_j^+) - \psi(\mathbf{c}_j)\right] + m(\mathbf{b})\psi(\mathbf{c}_j). \quad (4.6)\end{aligned}$$

The $k-$means clustering algorithm we apply to an initial partition of $\mathcal{B}$ is identical to the one provided in [19] and is briefly described below.

Algorithm 4.1 *$k-$means for BIRCH generated partitions.*

```
1. run iterations of batch $k-$means as long as new partitions are generated.

2. apply one iteration of incremental $k-$means.
   if (new partition is generated)
           go to step 1.

3. stop.
```

In Section 5 we report numerical experiment for clustering with the distance like function given by (3.4). We now display the right hand side of (4.6) for two special choices $\psi(\mathbf{x}) = \|\mathbf{x}\|^2$, and $\psi(\mathbf{x}) = \sum_{j=1}^{n} \mathbf{x}[j]\log \mathbf{x}[j] - \mathbf{x}[j]$. For convenience of presentation for a cluster $\pi_i^{\mathcal{B}}$ we denote $\sum_{\mathbf{b}\in\pi_i^{\mathcal{B}}} m(\mathbf{b})$ by M_i. For a vector $\mathbf{x} \in \mathbf{R}_+^n$ we denote $\sum_{j=1}^{n} \mathbf{x}[j]\log \mathbf{x}[j]$ by $\mathbf{x}^L\mathbf{x}$, and $\sum_{j=1}^{n} \mathbf{x}[j]$ by $\mathbf{x}^T\mathbf{e}$.

kernel $\psi(\mathbf{x})$	right–hand side of (4.6)
$\|\mathbf{x}\|^2$	$\frac{M_i \cdot m(\mathbf{b})}{M_i - m(\mathbf{b})} \|\mathbf{c}_i - \mathbf{b}\|^2 -$ $\frac{M_j \cdot m(\mathbf{b})}{M_j + m(\mathbf{b})} \|\mathbf{c}_j - \mathbf{b}\|^2$
$\mathbf{x}^L\mathbf{x} - \mathbf{x}^T\mathbf{e}$	$[M_i - m(\mathbf{b})]\left(\mathbf{c}_i^-\right)^L \mathbf{c}_i^- - M_i\mathbf{c}_i^L\mathbf{c}_i$ $+$ $[M_j + m(\mathbf{b})]\left(\mathbf{c}_j^+\right)^L \mathbf{c}_j^+ - M_j\mathbf{c}_j^L\mathbf{c}_j$

5. Numerical experiments

Results of preliminary numerical experiments with two datasets are reported in this section. In all the experiments n "best" terms are selected (see [19] for the selection procedure) to create the vector space model (see [2]), and the `tfn` is applied to normalize the vectors (see [8]). The three step clustering procedure is applied to each document collection with three values $(2,0)$, $(0,1)$, and $(20,1)$ for the non–negative parameters (ν,μ).

First we work with the three collections Medlars, CISI and Cranfield (classic3, total of 3891 documents available from `http://www.cs.utk.edu/~lsi/`). For the experiments with these three collection we select $n = 600$, $L = 5$, and $R = 5 \times 10^{-4} \times Q(\mathcal{A})$ (note that $Q(\mathcal{A})$ depends on the choice of the parameters (ν,μ)). The sparsity of the original dataset $\mathcal{A}$ is $\frac{\#\texttt{non}-\texttt{zero entries}}{\texttt{dim}\times\#\texttt{vectors}} \approx 4\%$, the corresponding sparsity for the constructed dataset $\mathcal{B}$ along with the average cluster size and the number of clusters are given in Table 2. Next PDDP generates 3 cluster initial partition of the

(ν, μ)	$(2,0)$	$(0,1)$	$(20,1)$
sparsity	12%	16%	14%
av clus size	3.11	4.41	3.73
# of clus	1249	881	1043

Table 2: sparsity, average cluster size and number of clusters generated by BIRCH with (ν, μ) distance for 3981 vectors of dimension 600

three datasets $\mathcal{B}$ generated by BIRCH. Batch $k-$means, and $k-$means are applied to the three initial partitions. Quality of final partitions of the original dataset $\mathcal{A}$ along with number of iterations performed by the $k-$means algorithms while clustering the datasets $\mathcal{B}$ are reported in Table 3.

(ν, μ)	$(2,0)$	$(0,1)$	$(20,1)$
PDDP	3612	44994	81087
batch $k-$means	3610 `it` $= 2$	44893 `it` $= 1$	81016 `it` $= 2$
$k-$means	3607 `it` $= 17$	44699 `it` $= 11$	80711 `it` $= 14$

Table 3: (ν, μ) quality of the dataset $\mathcal{A}$ (dimension 600, number of vectors 3981) partitions generated by PDDP, batch $k-$means, and $k-$means applied to the "squashed" dataset $\mathcal{B}$ produced by BIRCH (`it` indicates the number of iterations when appropriate)

Next consider the 20 newsgroups dataset of 19997 messages from 20 Usenet newsgroups. For the experiments with this collection we select $n = 1000$, $L = 10$, and $R = 5 \times 10^{-3} \times Q(\mathcal{A})$. The sparsity of the original dataset $\mathcal{A}$ is about 5%, the corresponding sparsity for the constructed dataset $\mathcal{B}$ along with average cluster size and number of clusters are given in Table 4. We use PDDP to generate the initial 20 cluster partitions for the three datasets $\mathcal{B}$ generated by BIRCH. Batch $k-$means, and $k-$means to are applied to these partitions and the clustering results are shown in Table 5.

(ν, μ)	$(2,0)$	$(0,1)$	$(20,1)$
sparsity	25%	25%	25%
av clus size	9.99	9.99	9.99
# of clus	2000	2000	2000

Table 4: sparsity, average cluster size and number of clusters generated by BIRCH with (ν, μ) distance for 19997 vectors of dimension 1000

(ν, μ)	$(2,0)$	$(0,1)$	$(20,1)$
PDDP	18065	268376	449026
batch $k-$means	17982 `it` $= 3$	267200 `it` $= 2$	447829 `it` $= 4$
$k-$means	17947 `it` $= 61$	265301 `it` $= 89$	444780 `it` $= 116$

Table 5: (ν, μ) quality of the dataset $\mathcal{A}$ (dimension 1000, number of vectors 19997) partitions generated by PDDP, batch $k-$means, and $k-$means applied to the "squashed" dataset $\mathcal{B}$ produced by BIRCH (`it` indicates the number of iterations when appropriate)

Table 6 presents results obtain by direct application of batch $k-$means, and $k-$means to initial partition generated by PDDP for classic3 collection.

Finally results pertaining to partitions generated by PDDP, batch $k-$means, and $k-$means from the 20 newsgroups dataset are reported in Table 7.

6. Conclusion

This paper extends the earlier work [22], [4] and suggests BIRCH and $k-$means clustering algo-

(ν, μ)	$(2,0)$	$(0,1)$	$(20,1)$
batch	3608	43759	80265
$k-$means	`it` $= 3$	`it` $= 4$	`it` $= 0$
	3605	43464	79520
$k-$means	`it` $= 87$	`it` $= 100$	`it` $= 246$

Table 6: (ν, μ) quality of the dataset $\mathcal{A}$ (dimension 600, number of vectors 3981) partitions generated by PDDP, batch $k-$means, and $k-$means (`it` indicates the number of iterations)

(ν, μ)	$(2,0)$	$(0,1)$	$(20,1)$
batch	17956	256472	443427
$k-$means	`it` $= 47$	`it` $= 18$	`it` $= 1$
	17808	250356	429921
$k-$means	`it` $= 5862$	`it` $= 7737$	`it` $= 9988$

Table 7: (ν, μ) quality of the dataset $\mathcal{A}$ (dimension 1000, number of vectors 19997) partitions generated by PDDP, batch $k-$means, and $k-$means (`it` indicates the number of iterations)

rithms with Bregman divergences. Numerical experiment with three specific distance like functions (the squared Euclidean distance, the Kullback–Leibler divergence, and a positive linear combination of both) are provided. The experiments show a trade off between the running time (number of iterations) and quality of the obtained partitions. A good choice of the constants L and R for the BIRCH part of the procedure is important for good data "squashing." For example in the quadratic case the choice of $L = 10$ and $R = 5 \times 10^{-3} \times Q(\mathcal{A})$ selected for the experiments with 20 newsgroups dataset allows BIRCH to generate clusters with up to 10 vector with sample variance 100 times as much as the sample variance of the entire dataset, i.e. in this case for all practical matters cluster size is the only criterion governing cluster building process by BIRCH. As a result we get 2000 clusters of size 10 (see Table 4). With $L = 10$ and $R = 5 \times 10^{-5} \times Q(\mathcal{A})$ "quadratic" BIRCH generates 9469 clusters, the dataset $\mathcal{B}$ sparsity is about 8%, and the average cluster size is 2.11 (with max cluster size 8, and min cluster size 2). Result of clustering this dataset are reported in Table 8.

(ν, μ)	$(2,0)$
PDDP	18195
batch	18056
$k-$means	`it` $= 16$
	17924
$k-$means	`it` $= 1250$

Table 8: (ν, μ) quality of the dataset $\mathcal{A}$ (dimension 1000, number of vectors 19997) partition generated by PDDP, quadratic batch $k-$means, and quadratic $k-$means through a BIRCH generated partition with 9469 clusters (`it` indicates the number of iterations)

While PDDP does an excellent job as an "initial partition generator" it is of interest to devise "initial partition generators" that reflect the nature of a distance like function used by BIRCH and $k-$means.

It is of interest to investigate efficiency of BIRCH combined with $k-$means smoothing techniques recently proposed in [20]. Usefulness of this approach will be tested on large datasets like, for example, the Enron dataset.

References

[1] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman diver-

gences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[2] M. Berry and M. Browne. *Understanding Search Engines.* SIAM, 1999.

[3] D. L. Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, 1998.

[4] Paul S. Bradley, Usama M. Fayyad, and Cory Reina. Scaling clustering algorithms to large databases. In *Knowledge Discovery and Data Mining*, pages 9–15, 1998.

[5] L.M. Bregman. A relaxation method of finding a common point of convex sets and its application to the solution of problems in convex programming. *USSR Comp. Math. and Math Phys.*, 7:200–217, 1967.

[6] Y. Censor and A. Lent. An interval row action method for interval convex programming. *J. of Optimization Theory and Applications*, 34:321–353, 1981.

[7] Y. Censor and S.A. Zenios. *Parallel Optimization.* Oxford University Press, Oxford, 1997.

[8] E. Chisholm and T. Kolda. New term weighting formulas for the vector space method in information retrieval, 1999. Report ORNL/TM-13756, Computer Science and Mathematics Division, Oak Ridge National Laboratory.

[9] M. Collins, R.E. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. In *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 158 – 169, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[10] M. Collins, R.E. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48:253–285, 2002.

[11] I. Csiszar. Information-type measures of difference of probability distributions and indirect observations. *Studia Sci. Mat. Hungar.*, 2:299–318, 1967.

[12] J. Kogan, M. Teboulle, and C. Nicholas. The entropic geometric means algorithm: an approach for building small clusters for large text datasets. In D. Boley et al, editor, *Proceedings of the Workshop on Clustering Large Data Sets (held in conjunction with the Third IEEE International Conference on Data Mining)*, pages 63–71, 2003.

[13] J. Kogan, M. Teboulle, and C. Nicholas. Data driven similarity measures for k–means like clustering algorithms. *Information Retrieval*, 8:331–349, 2005.

[14] J. Lafferty. Adaptive models, boosting and inference for generalized divergences. In *Proceedings of 12th Annual Conference on Computational Learning Theory*, pages 125–133, 1999.

[15] J. Lafferty, S.D. Pietra, and Pietra V.D. Statistical learning algorithms based on Bregman distances. In *Proceedings of the Canadian Workshop on Information Theory*, 1997.

[16] M. Teboulle. Entropic proximal mappings with application to nonlinear programming. *Mathematics of Operation Research*, 17:670–690, 1992.

[17] M. Teboulle. On φ-divergence and its applications. In F.Y. Phillips and J. Rousseau, editors, *Systems and Management Science by Extremal Methods–Research Honoring Abraham Charnes at Age 70*, pages 255–273, Kluwer Academic Publishers. Nowell, MA, 1992.

[18] M. Teboulle. Convergence of proximal-like algorithms. *SIAM J. of Optimization*, 7:1069–1083, 1997.

[19] M. Teboulle, P. Berkhin, I. Dhillon, Y. Guan, and J. Kogan. Clustering with entropy–like k–means algorithms. In J. Kogan, C. Nicholas, and M. Teboulle, editors, *Grouping Multidimensional Data: Recent Advances in Clustering*, pages 127–160. Springer–Verlag, 2006.

[20] M. Teboulle and J. Kogan. Deterministic annealing and a k-means type smoothing optimization algorithm for data clustering. In I. Dhillon, J. Ghosh, and J. Kogan, editors, *Proceedings of the Workshop on Clustering High Dimensional Data and its Applications (held in conjunction with the Fifth SIAM International Conference on Data Mining)*, pages 13–22, Philadelphia, PA, 2005. SIAM.

[21] S. Wang and D. Schuurmans. Learning continuous latent variable models with Bregman divergences. In *Lecture Notes in Artificial Intelligence*, volume 2842, pages 190–204, 2003.

[22] T. Zhang, R. Ramakrishnan, and M. Livny. BIRCH: A new data clustering algorithm and its applications. *Journal of Data Mining and Knowledge Discovery*, 1(2):141–182, 1997.