
Strong Coresets for Hard and Soft Bregman Clustering with Applications to Exponential Family Mixtures

Mario Lucic[†]
ETH Zurich

Olivier Bachem[†]
ETH Zurich

Andreas Krause
ETH Zurich

Abstract

Coresets are efficient representations of data sets such that models trained on the coreset are *provably* competitive with models trained on the original data set. As such, they have been successfully used to scale up clustering models such as K-Means and Gaussian mixture models to massive data sets. However, until now, the algorithms and the corresponding theory were usually specific to each clustering problem.

We propose a single, practical algorithm to construct strong coresets for a large class of hard and soft clustering problems based on Bregman divergences. This class includes hard clustering with popular distortion measures such as the Squared Euclidean distance, the Mahalanobis distance, KL-divergence and Itakura-Saito distance. The corresponding soft clustering problems are directly related to popular mixture models due to a dual relationship between Bregman divergences and Exponential family distributions. Our theoretical results further imply a randomized polynomial-time approximation scheme for hard clustering. We demonstrate the practicality of the proposed algorithm in an empirical evaluation.

1 INTRODUCTION

Clustering is the task of partitioning data points into groups (or clusters) such that similar data points are assigned to the same cluster. It is widely used in machine learning, data mining and information theory

and remains an important unsupervised learning problem: According to Wu et al. (2008), *Lloyd's algorithm*, a local search algorithm for K-Means, is still one of the ten most popular algorithms for data mining.

Over the last several years, the world has witnessed the emergence of data sets of an unprecedented scale across different scientific disciplines. The large size of such data sets presents new computational challenges as existing algorithms can be computationally infeasible in the context of millions or even billions of data points. Clustering is not exempt from this problem: Popular algorithms to solve both hard clustering problems (such as K-Means) and soft clustering problems (such as *Gaussian mixture models*) typically require multiple passes through a data set.

A well established technique for scaling several clustering problems (including K-Means and GMMs) is based on *coresets* – a data summarization technique originating from computational geometry. A *coreset* is an efficient representation of the full data set such that models trained on a coreset are *provably* competitive with models trained on the original data set. Since coresets are typically small and easy to construct, they allow for efficient approximate inference with strong theoretical guarantees. However, until now, the algorithms and the corresponding theory were specific to each clustering problem.

Bregman divergences are a class of dissimilarity measures that are characterized by the property that the mean of a set of points is the optimal representative of that set. As such, the *Bregman hard clustering problem* generalizes a variety of important clustering problems such as K-Means and various information theoretic extensions (Banerjee et al., 2005). A family of corresponding soft clustering problems are denoted by *Bregman soft clustering* (Banerjee et al., 2005) and are closely related to fitting mixture models with exponential family distributions. Hence, *Bregman clustering* offers a natural framework for studying a variety of hard and soft clustering problems.

Appearing in Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 51. Copyright 2016 by the authors.

[†]These authors contributed equally to this work.

DOMAIN	$\phi(t)$	μ	A	$d_\phi(p, q)$	DIVERGENCE
\mathbb{R}^d	$\ t\ _2^2$	1	I_d	$\ p - q\ _2^2$	Squared Euclidean
\mathbb{R}^d	$t^T A t$	1	A	$(p - q)^T A (p - q)$	Mahalanobis
$[\lambda, \nu]^d \subseteq \mathbb{R}_+^d$	$\sum t_i \ln(t_i) - t_i$	$\frac{\lambda}{\nu^2}$	$\frac{1}{2\lambda} I_d$	$\sum p_i \ln p_i / q_i - \sum (p_i - q_i)$	Relative Entropy
$[\lambda, \nu]^d \subseteq \mathbb{R}_+^d$	$-\sum \ln(t_i)$	$\frac{\lambda^2}{\nu^2}$	$\frac{1}{2\lambda^2} I_d$	$\sum p_i / q_i - \ln(p_i / q_i) - 1$	Itakura-Saito
$[\lambda, \nu]^d \subseteq \mathbb{R}_+^d$	$\sum 1/t_i^\alpha$	$\frac{\lambda^{\alpha+2}}{\nu^{\alpha+2}}$	$\frac{\alpha(1-\alpha)}{2\lambda^{\alpha+2}} I_d$	$\sum 1/p_i^\alpha - (\alpha + 1)/q_i^\alpha + \alpha p_i / q_i^{\alpha+1}$	Harmonic ($\alpha > 0$)
$[\lambda, \nu]^d \subseteq \mathbb{R}_+^d$	$\sum t_i^\alpha$	$\frac{\lambda^{\alpha-2}}{\nu^{\alpha-2}}$	$\frac{\alpha(1-\alpha)}{2} \nu^{\alpha-2} I_d$	$\sum p_i^\alpha + (\alpha - 1)q_i^\alpha - \alpha p_i q_i^{\alpha-1}$	Norm-like ($\alpha > 2$)
$[\lambda, \nu]^d \subseteq \mathbb{R}_+^d$	$\sum e^{t_i}$	$e^{-(\nu-\lambda)}$	$\frac{e^\nu}{2} I_d$	$\sum e^{p_i} - (p_i - q_i + 1)e^{q_i}$	Exponential Loss
$[-\nu, \nu]^d \subseteq (-1, 1)^d$	$-\sum \sqrt{1 - t_i^2}$	$2(1 - \nu^2)^{3/2}$	$\frac{1}{2(1-\nu)^{3/2}} I_d$	$\sum (1 - p_i q_i) / (\sqrt{1 - q_i^2}) - \sqrt{1 - p_i^2}$	Hellinger

 Table 1: Selection of μ -similar Bregman divergences (Ackermann and Blömer, 2009).

Our contributions. In this paper, we consider the problem of scaling Bregman clustering to massive data sets using coresets. As our key contributions, we:

- provide strong coresets of size *independent of the data set size* for all μ -similar Bregman divergences,
- prove that the *same* practical coreset construction works for both hard and soft clustering problems,
- provide a randomized polynomial-time approximation scheme for the corresponding hard clustering problems,
- establish the combinatorial complexity of mixture models induced by the regular exponential family,
- demonstrate the practicality of the proposed algorithm in an empirical evaluation.

2 BACKGROUND

Bregman divergence. For any strictly convex, differentiable function $\phi : \mathcal{K} \rightarrow \mathbb{R}$, the Bregman divergence with respect to ϕ for all $p, q \in \mathcal{K}$ is defined as

$$d_\phi(p, q) = \phi(p) - \phi(q) - \nabla \phi(q)^T (p - q).$$

For example, setting $\mathcal{K} = \mathbb{R}^d$ and $\phi(q) = \|q\|_2^2$ results in $d_\phi(p, q) = \|p - q\|_2^2$ which is the squared Euclidean distance. Bregman divergences are characterized by the fact that the mean of a set of points minimizes the sum of Bregman divergences between these points and any other point. More formally, for any Bregman divergence d_ϕ and any finite set $\mathcal{X} \subseteq \mathcal{K}$, it holds that

$$\arg \min_{z \in \mathbb{R}^d} \sum_{x \in \mathcal{X}} d_\phi(x, z) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} x. \quad (1)$$

Furthermore, Banerjee et al. (2005) proved that any function satisfying (1) is a Bregman divergence.

μ -similar Bregman divergences are a subclass of Bregman divergences related to the squared Mahalanobis distance.¹ The class of μ -similar Bregman divergences

¹The squared Mahalanobis distance d_A for any positive definite matrix $A \in \mathbb{R}^{d \times d}$ and any $p, q \in \mathbb{R}^d$ is defined as $d_A(p, q) = (p - q)^T A (p - q)$.

includes many popular Bregman divergences such as the squared Euclidean distance, the squared Mahalanobis distance, the KL-divergence and the Itakura-Saito distance. Several other divergences from this class are shown in Table 1.

Definition 1 (μ -similar Bregman divergence)

A Bregman divergence d_ϕ on domain $\mathcal{K} \subseteq \mathbb{R}^d$ is μ -similar for some $\mu > 0$ iff there exists a positive definite matrix $A \in \mathbb{R}^{d \times d}$ such that, for each $p, q \in \mathcal{K}$,

$$\mu d_A(p, q) \leq d_\phi(p, q) \leq d_A(p, q)$$

where d_A denotes the squared Mahalanobis distance.

Coresets. A coreset is a weighted subset of the data such that the quality of any clustering evaluated on the coreset closely approximates the quality on the full data set. Consider a cost function that depends on a set of points \mathcal{X} and a query $Q \in \mathcal{Q}$ that is additively decomposable into non-negative functions $\{f_Q(x)\}_{x \in \mathcal{X}}$, i.e.

$$\text{cost}(\mathcal{X}, Q) = \sum_{x \in \mathcal{X}} f_Q(x).$$

For example, for the K-Means clustering problem, $f_Q(x)$ is the squared Euclidean distance of x to the closest cluster center in the set Q . The key idea behind coresets is to find a weighted subset \mathcal{C} such that the cost of a query Q can be approximated on \mathcal{C} by

$$\text{cost}(\mathcal{C}, Q) = \sum_{(w, c) \in \mathcal{C}} w f_Q(c).$$

A weighted subset \mathcal{C} is an ε -coreset of \mathcal{X} if it approximates the cost function of the full data set up to a multiplicative factor of $1 \pm \varepsilon$ uniformly for *all* queries $Q \in \mathcal{Q}$, i.e.

$$(1 - \varepsilon) \text{cost}(\mathcal{X}, Q) \leq \text{cost}(\mathcal{C}, Q) \leq (1 + \varepsilon) \text{cost}(\mathcal{X}, Q).$$

Since the cost contributions $f_Q(x)$ and the space of queries \mathcal{Q} depend on the problem at hand, coresets are inherently problem-specific.

Coresets have been the subject of several recent papers, with focus on unsupervised parametric and non-parametric models (Feldman et al., 2011; Balcan et al., 2013; Bachem et al., 2015; Lucic et al., 2015), as well as in the context of empirical risk minimization (Reddi et al., 2015).

As the coreset property bounds the approximation error for all queries, the difference between the solution on the full data set and the solution on the coreset is bounded. Hence, one can use any solver on the coreset instead of the full data set and obtain provable approximation guarantees.² At the same time, coresets are usually sublinear in (or even independent of) the number of samples which implies that one can run computationally intensive algorithms that would otherwise be infeasible.

Additionally, coresets are a practical and flexible tool that requires no assumptions on the data. While the theory behind coresets requires elaborate tools from computational geometry, the resulting coreset construction algorithms are simple to implement.

A key property of coresets is that they can be constructed both in a distributed and a streaming setting. The constructions rely on the property that both unions of coresets and coresets of coresets are coresets (Har-Peled and Mazumdar, 2004). In fact, Feldman et al. (2013) use these properties to construct coresets in a tree-wise fashion which can be parallelized in a Map-Reduce style or used to maintain an up-to-date coreset in a streaming setting by applying the *static-to-dynamic transformation* (Bentley and Saxe, 1980).

3 STRONG CORESETS FOR HARD CLUSTERING

The goal of the K-Means clustering problem is to find k cluster centers C such that the quantization error

$$\text{cost}_{\text{kmeans}}(\mathcal{X}, C) = \sum_{x \in \mathcal{X}} \min_{c \in C} \|x - c\|_2^2$$

is minimized, where $\mathcal{X} \subseteq \mathbb{R}^d$ denotes the set of data points to be clustered. By replacing the squared Euclidean distance $\|\cdot\|_2^2$ with a Bregman divergence d_ϕ , K-Means clustering generalizes to *Bregman hard clustering*. In this problem the goal is to compute a set of k cluster centers C minimizing

$$\text{cost}_H(\mathcal{X}, C) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} d_\phi(x, C) \quad (2)$$

²To account for weighted data, solvers can either be naturally extended (e.g. Algorithms 1 and 5), or weighted data points can be replaced by multiple copies of the same point after appropriate rescaling.

Algorithm 1 Bregman hard k -clustering with d_ϕ

Require: \mathcal{X}, k , initial centers $\{\mu_j\}_{j=1}^k$

```

1: repeat
2:   for  $j \leftarrow 1$  to  $k$ 
3:      $\mathcal{X}_j \leftarrow \emptyset$ 
4:   for  $i \leftarrow 1$  to  $n$ 
5:      $j \leftarrow \arg \min_{1 \leq \ell \leq k} d_\phi(x_i, \mu_\ell)$ 
6:      $\mathcal{X}_j \leftarrow \mathcal{X}_j \cup \{x_i\}$ 
7:   for  $j \leftarrow 1$  to  $k$ 
8:      $\mu_j \leftarrow \frac{1}{|\mathcal{X}_j|} \sum_{x \in \mathcal{X}_j} x$ 
9: until convergence
10: return  $\{\mu_j\}_{j=1}^k$ 

```

where we define $d_\phi(x, C) = \min_{c \in C} d_\phi(x, c)$. For weighted data sets, the contribution of a point to the cost function is scaled by its weight $\nu(x)$.

The notion of hard clustering relates to the fact that, as in K-Means clustering, the *minimum* with regards to the set of cluster centers C leads to a hard assignment of each data point to its closest cluster center. Furthermore, the assignment boundary between two arbitrary centers is a $(d - 1)$ -dimensional hyperplane and the set of cluster centers C induces a Voronoi partitioning on the data set \mathcal{X} (Banerjee et al., 2005).

Lloyd’s algorithm (Lloyd, 1982) solves the K-Means problem by iterating between assigning points to closest cluster centers and recalculating the centers as the mean of the assigned points. As the mean is an optimal representative of a single cluster for any Bregman divergence, Lloyd’s algorithm can be naturally generalized to the *Bregman hard clustering algorithm* detailed in Algorithm 1 (Banerjee et al., 2005).

Each iteration of Algorithm 1 incurs a computational cost of $\mathcal{O}(nkd)$. As the number of iterations until convergence can be very large, coresets can be used to scale Bregman hard clustering. Previous approaches either only considered *weak* coresets (Ackermann and Blömer, 2009) or impose prior assumptions on the data (Feldman et al., 2013). In contrast, we provide a significantly stronger theoretical guarantee via *strong* coresets, i.e. coresets for which the approximation guarantee holds for *all* possible queries.

Definition 2 (Coreset definition) Let $\varepsilon > 0$ and $k \in \mathbb{N}$. Let d_ϕ be a μ -similar Bregman divergence on domain \mathcal{K} and $\mathcal{X} \subseteq \mathcal{K}$ be a finite set of points. The weighted set \mathcal{C} is an (ε, k) coreset of \mathcal{X} for hard clustering with d_ϕ if for any set $Q \subseteq \mathcal{K}$ of cardinality k

$$|\text{cost}_H(\mathcal{X}, Q) - \text{cost}_H(\mathcal{C}, Q)| \leq \varepsilon \text{cost}_H(\mathcal{X}, Q).$$

3.1 Coreset construction algorithm

The idea behind the proposed coreset construction is straightforward: The objective function for Bregman hard clustering in (2) is additively decomposable over \mathcal{X} and the contribution of each point is independent of all the other points. Hence, the cost evaluated on a uniform subsample of the data points is an unbiased estimator of the true objective function.

While this intuition is simple, unbiasedness is not sufficient and uniform subsampling does not provide the strong theoretical guarantee required by Definition 2: Single points can potentially have a large impact on the objective function and force the sample size to $\Omega(n)$.³ In particular, this occurs if the clusters are imbalanced or if there are points far away from the bulk of the data. Moreover, the coreset property in Definition 2 needs to hold uniformly for all queries in the (possibly infinite) set Q .

To obtain the strong coreset property, we propose a coreset construction based on the *importance sampling* framework by Langberg and Schulman (2010) and Feldman and Langberg (2011). Consider a Bregman hard clustering problem defined by a μ -similar Bregman divergence d_ϕ on domain \mathcal{K} and a finite data set $\mathcal{X} \subseteq \mathcal{K}$. The construction consists of two steps:

Step 1. We first find a rough approximation (*bicriteria approximation*) of the optimal clustering. As μ -similar Bregman divergences are closely related to the squared Mahalanobis distance, we show in the proof of Theorem 1 that it is sufficient to find a rough approximation with regards to the Mahalanobis distance. To this end, we apply D^2 -sampling which was previously analyzed by Arthur and Vassilvitskii (2007) in the context of the popular **k-means++** algorithm. The idea is to sample data points as cluster centers using an adaptive sampling scheme: the first cluster center is sampled uniformly at random and additional points are then iteratively sampled with probability proportional to the minimum squared Mahalanobis distance to the already sampled cluster centers. We prove that this procedure detailed in Algorithm 2 produces a solution that is, with constant probability, $\mathcal{O}(\log k)$ competitive with the optimal clustering in terms of Mahalanobis distance. Under natural assumptions on the data, a bicriteria approximation can even be computed in sublinear time (Bachem et al., 2016).

Step 2. The rough approximation is then used in Algorithm 3 to compute an importance sampling distribution. The idea is to sample points with a potentially

³A simple example is a data set where a first cluster contains $n - 1$ points at a single location and a second cluster consists of one point arbitrarily far away from the first cluster.

Algorithm 2 Mahalanobis D^2 -sampling

Require: \mathcal{X}, k, d_A

- 1: Uniformly sample $x \in \mathcal{X}$ and set $B = \{x\}$.
 - 2: **for** $i \leftarrow 2, 3, \dots, k$
 - 3: Sample $x \in \mathcal{X}$ with probability $\frac{d_A(x, B)}{\sum_{x' \in \mathcal{X}} d_A(x', B)}$ and add it to B .
 - 4: **return** B
-

Algorithm 3 Coreset construction

Require: $\mathcal{X}, k, B, m, d_A$

- 1: $\alpha \leftarrow 16(\log k + 2)$
 - 2: **for each** b_i in B
 - 3: $B_i \leftarrow$ Set of points from \mathcal{X} closest to b_i in terms of d_A . Ties broken arbitrarily.
 - 4: $c_\phi \leftarrow \frac{1}{|\mathcal{X}|} \sum_{x' \in \mathcal{X}} d_A(x', B)$
 - 5: **for each** $b_i \in B$ and $x \in B_i$
 - 6: $s(x) \leftarrow \frac{\alpha d_A(x, B)}{c_\phi} + \frac{2\alpha \sum_{x' \in B_i} d_A(x', B)}{|B_i| c_\phi} + \frac{4|\mathcal{X}|}{|B_i|}$
 - 7: **for each** $x \in \mathcal{X}$
 - 8: $p(x) \leftarrow s(x) / \sum_{x' \in \mathcal{X}} s(x')$
 - 9: $\mathcal{C} \leftarrow$ Sample m weighted points from \mathcal{X} where each point x has weight $\frac{1}{mp(x)}$ and is sampled with probability $p(x)$.
 - 10: **return** \mathcal{C}
-

high impact on the objective more frequently but assign them a lower weight. The sensitivity $s(x)$ of a point $x \in \mathcal{X}$ is the maximal ratio between the cost contribution of that point and the average contribution of all points (Langberg and Schulman, 2010), i.e.

$$\sigma(x) = \max_{Q \subseteq \mathcal{K}: |Q|=k} \frac{d_\phi(x, Q)}{\frac{1}{|\mathcal{X}|} \sum_{x' \in \mathcal{X}} d_\phi(x', Q)}.$$

We derive an upper bound for $\sigma(x)$ and use it as the sampling distribution in Algorithm 3. Intuitively, this specific choice bounds the variance of the importance sampling scheme (Feldman and Langberg, 2011) and, if we sample enough points, produces a coreset. This result is formally stated and proven in Theorem 1 where we provide a bound on the required coreset size.

Finally, we can solve the Bregman hard clustering problem on the coreset using a weighted version of the Bregman hard clustering algorithm.

3.2 Analysis

Our main result is that this construction leads to valid coresets for the Bregman hard clustering problem. In particular, the required coreset size does not depend on the size n of the original data set.

Theorem 1 *Let $\varepsilon \in (0, 1/4)$, $\delta > 0$ and $k \in \mathbb{N}$. Let d_ϕ be a μ -similar Bregman divergence on domain \mathcal{K} and denote by d_A the corresponding squared Mahalanobis*

distance. Let \mathcal{X} be a set of points in \mathcal{K} and let $B \subseteq \mathcal{X}$ be the set with the smallest quantization error in terms of d_A among $\mathcal{O}(\log \frac{1}{\delta})$ runs of Algorithm 2. Let C be the output of Algorithm 3 with

$$m = \mathcal{O}\left(\frac{dk^3 + k^2 \log \frac{1}{\delta}}{\mu^2 \varepsilon^2}\right).$$

Then, with probability at least $1 - \delta$, the set C is a (ε, k) -coreset of \mathcal{X} for hard clustering with d_ϕ .

The proof is provided in Section C of the supplementary material. The main steps include bounding the sensitivity in Lemma 2 and bounding the combinatorial complexity of Bregman hard clustering in Theorem 6. In practice, it is usually sufficient to run Algorithm 2 only once and to fix the coreset size m instead of ε (see Section 5).

As an immediate consequence of the coreset property, the optimal clustering obtained on the coreset is provably competitive with the optimal clustering on the full data set when evaluated on the full data set.

Lemma 1 *Let $\varepsilon \in (0, 1)$ and let d_ϕ be a μ -similar Bregman divergence on domain \mathcal{K} . Let $\mathcal{X} \subseteq \mathcal{K}$ be a data set, $k \in \mathbb{N}$ and C be an $(\varepsilon/3, k)$ -coreset of \mathcal{X} for hard clustering with d_ϕ . Let $Q_\mathcal{X}^*$ and Q_C^* denote the optimal set of cluster centers for \mathcal{X} and C respectively. Then,*

$$\text{cost}_H(\mathcal{X}, Q_C^*) \leq (1 + \varepsilon) \text{cost}_H(\mathcal{X}, Q_\mathcal{X}^*).$$

The proof is presented in the Section C of the supplementary material.

3.3 Randomized polynomial-time approximation scheme

The fact that the size of the proposed coresets is independent of the number of data points n readily implies a randomized polynomial-time approximation scheme (PTAS) for Bregman hard clustering with μ -similar Bregman divergences. We first generate a coreset using Algorithm 3 and then consider all possible k partitionings of the coreset points. By the coreset property, it is guaranteed that the centers of the best partitioning are $1 + \varepsilon$ competitive with the optimal solution.

Theorem 2 *Let $\varepsilon \in (0, 3/4)$, $\delta > 0$ and let d_ϕ be a μ -similar Bregman divergence on domain \mathcal{K} . Let $\mathcal{X} \subseteq \mathcal{K}$ be a data set, $k \in \mathbb{N}$ and ε the desired approximation error. Let Q^* be the best solution from $\mathcal{O}(\log \frac{1}{\delta})$ runs of Algorithm 4. Then, with probability at least $1 - \delta$,*

$$\text{cost}_H(\mathcal{X}, Q^*) \leq (1 + \varepsilon) \min_Q \text{cost}_H(\mathcal{X}, Q).$$

The time complexity is $\mathcal{O}((nkd + 2^{\text{poly}(kd/\mu\varepsilon)}) \log \frac{1}{\delta})$.

Algorithm 4 Randomized PTAS

Require: $\mathcal{X}, k, \varepsilon, d_\phi$

- 1: $C \leftarrow (k, \varepsilon/3)$ -coreset for \mathcal{X} with respect to d_ϕ .
 - 2: $\mathcal{P} \leftarrow$ Centers of all k -partitionings of C .
 - 3: $Q^* \leftarrow \arg \min_{P \in \mathcal{P}} \frac{1}{|C|} \sum_{(w, c) \in C} w d_\phi(c, P)$
 - 4: **return** Q^*
-

The correctness follows from Lemma 1 and the fact that the number of k partitionings of the coreset points is independent of n . As this algorithm is primarily of theoretical interest, we recommend running the approach presented in Section 3.1 to solve Bregman clustering problems in practice.

4 STRONG CORESETS FOR SOFT CLUSTERING

In hard clustering each data point is assigned to exactly one cluster. In contrast, in soft clustering each data point is assigned to each cluster with a certain probability. A prototypical example of soft clustering is fitting the parameters of a Gaussian mixture model in which one assumes that all data points are generated from a mixture of a finite number of Gaussians with unknown parameters. Other popular models include the Poisson mixture model, the mixture of multinomials and the mixture of exponentials.

Banerjee et al. (2005) show that there is a bijection between regular exponential family distributions and Bregman divergences. In particular, the log-likelihood of exponential family mixture models can be expressed in terms of Bregman divergences (see Section 4.2). By considering the resulting objective, one obtains *Bregman soft clustering*. The intuition is that Bregman hard clustering can be turned into a soft clustering problem by replacing the *min* function by a *soft-min* function. More formally, let d_ϕ be a Bregman divergence and $\mathcal{X} \subseteq \mathcal{K}$ be a set of n points. Let $k \in \mathbb{N}$, $w = (w_1, \dots, w_k) \subseteq \mathbb{R}^k$ and $\theta = (\theta_1, \dots, \theta_k) \subseteq \mathbb{R}^{kd}$ and let Q be the concatenation of w and θ . The goal of Bregman soft clustering is to minimize

$$\text{cost}_S(\mathcal{X}, Q) = - \sum_{i=1}^n \ln \left(\sum_{j=1}^k w_j \exp(-d_\phi(x_i, \theta_j)) \right)$$

with respect to Q under the constraint that $w_j > 0$, $1 \leq j \leq k$ and $\sum_{j=1}^k w_j = 1$. Similar to Bregman hard clustering, the soft clustering problem can be solved using an *expectation-maximization* algorithm (Banerjee et al., 2005) which is detailed in Algorithm 5. The main difference with respect to hard clustering is that a probability distribution over assignments of points to clusters is maintained.

Algorithm 5 Bregman soft clustering with d_ϕ

Require: \mathcal{X} , k , initial parameters $\{(\theta_j, w_j)\}_{j=1}^k$

```

1: repeat
2:   for  $i = 1$  to  $n$ 
3:     for  $j = 1$  to  $k$ 
4:        $\eta_{ij} = \frac{w_j \exp(-d_\phi(x_i, \theta_j))}{\sum_{\ell=1}^k w_\ell \exp(-d_\phi(x_i, \theta_\ell))}$ 
5:     for  $j = 1$  to  $k$ 
6:        $w_j \leftarrow \frac{1}{n} \sum_{i=1}^n \eta_{ij}$ 
7:        $\theta_j \leftarrow \frac{\sum_{i=1}^n \eta_{ij} x_i}{\sum_{i=1}^n \eta_{ij}}$ 
8:   until convergence
9: return  $\{\theta_j, w_j\}_{j=1}^k$ 

```

4.1 Coresets for Bregman soft clustering

As in the hard clustering case, we define the coreset property in terms of a cost function with respect to some set of queries. In the case of soft clustering, the queries are the parameters of the mixture model.

Definition 3 Let $\varepsilon > 0$ and $k \in \mathbb{N}$. Let d_ϕ be a μ -similar Bregman divergence on domain \mathcal{K} and $\mathcal{X} \subseteq \mathcal{K}$ be a set of points. The weighted set \mathcal{C} is an (ε, k) coreset of \mathcal{X} for soft clustering with d_ϕ if for any set $Q \subseteq \mathcal{K}$ of cardinality k

$$|\text{cost}_s(\mathcal{X}, Q) - \text{cost}_s(\mathcal{C}, Q)| \leq \varepsilon \text{cost}_s(\mathcal{X}, Q).$$

We prove that the same coreset construction provided in Section 3.1 for hard clustering also computes valid coresets for the soft clustering problem. However, due to the higher combinatorial complexity of the underlying function space, more points need to be sampled to guarantee that the resulting weighted set is a coreset.

Theorem 3 Let $\varepsilon \in (0, 1/4)$, $\delta > 0$ and $k \in \mathbb{N}$. Let d_ϕ be a μ -similar Bregman divergence on domain \mathcal{K} and denote by d_A the corresponding squared Mahalanobis distance. Let \mathcal{X} be a set of points in \mathcal{K} and let $B \subseteq \mathcal{X}$ be the set with the smallest quantization error in terms of d_A among $\mathcal{O}(\log \frac{1}{\delta})$ runs of Algorithm 2. Let \mathcal{C} be the output of Algorithm 3 with coreset size

$$m = \mathcal{O}\left(\frac{d^2 k^4 + k^2 \log \frac{1}{\delta}}{\mu^2 \varepsilon^2}\right).$$

Then, with probability at least $1 - \delta$, the set \mathcal{C} is a (ε, k) -coreset of \mathcal{X} for soft clustering with d_ϕ .

The proof which includes a bound on the combinatorial complexity of mixtures of regular exponential family distributions is provided in Section D of the supplementary material.

4.2 Estimation for exponential family mixtures

Finding the maximum likelihood estimate of a set of parameters for a single exponential family can be done efficiently through the use of sufficient statistics. However, fitting the parameters of a mixture is a notoriously hard task – the mixture is not in the exponential family. More formally, consider a set $\tilde{\mathcal{X}}$ of n points drawn independently from a stochastic source that is a mixture of k densities of the same exponential family. Given $\tilde{\mathcal{X}}$, we would like to estimate the parameters $\{w_j, \theta_j\}_{j=1}^k$ of the mixture model using maximum likelihood estimation, i.e.

$$\max_{\{w_j, \theta_j\}_{j=1}^k} \mathcal{L}(\tilde{\mathcal{X}} | \theta) = \sum_{i=1}^n \ln \left(\sum_{j=1}^k w_j p_\psi(\tilde{x}_i | \theta_j) \right). \quad (3)$$

As shown by Banerjee et al. (2005) there is a bijection between regular exponential families and regular Bregman divergences that allows us to rewrite (3) as

$$\begin{aligned} \mathcal{L}(\mathcal{X} | Q) &= \sum_{i=1}^n \ln \left(\sum_{j=1}^k w_j \exp(-d_\phi(x, \eta_j)) b_\phi(x) \right) \\ &= \sum_{i=1}^n \ln(b_\phi(x_i)) \\ &\quad + \sum_{i=1}^n \ln \left(\sum_{j=1}^k w_j \exp(-d_\phi(x_i, \eta_j)) \right) \end{aligned}$$

where d_ϕ is the corresponding Bregman divergence and both \tilde{x} and η_j are related to x and θ_j by Legendre duality via ϕ . Since the first summand is independent of the model parameters, maximizing the likelihood of the mixture model is equivalent to minimizing

$$\text{cost}_s(\mathcal{X}, Q) = - \sum_{i=1}^n \ln \left(\sum_{j=1}^k w_j \exp(-d_\phi(x_i, \theta_j)) \right).$$

This is precisely the cost function of the coreset defined in Section 4.1. Hence, our coreset construction can be used for maximum likelihood estimation of regular exponential family mixtures by first constructing a coreset and then applying Bregman soft clustering (Algorithm 5) on the coreset. Moreover, as $\varepsilon \rightarrow 0$

$$|\mathcal{L}(\mathcal{X} | Q) - \mathcal{L}(\mathcal{C} | Q)| = |\text{cost}_s(\mathcal{X}, Q) - \text{cost}_s(\mathcal{C}, Q)| \rightarrow 0$$

uniformly over $Q \in \mathcal{Q}$. As a byproduct of our theoretical analysis, we also provide a bound of $\mathcal{O}(k^4 d^2)$ on the combinatorial complexity of mixtures of regular exponential family distributions with k components and d dimensions (see Theorem 7 in Section D of the supplementary material).

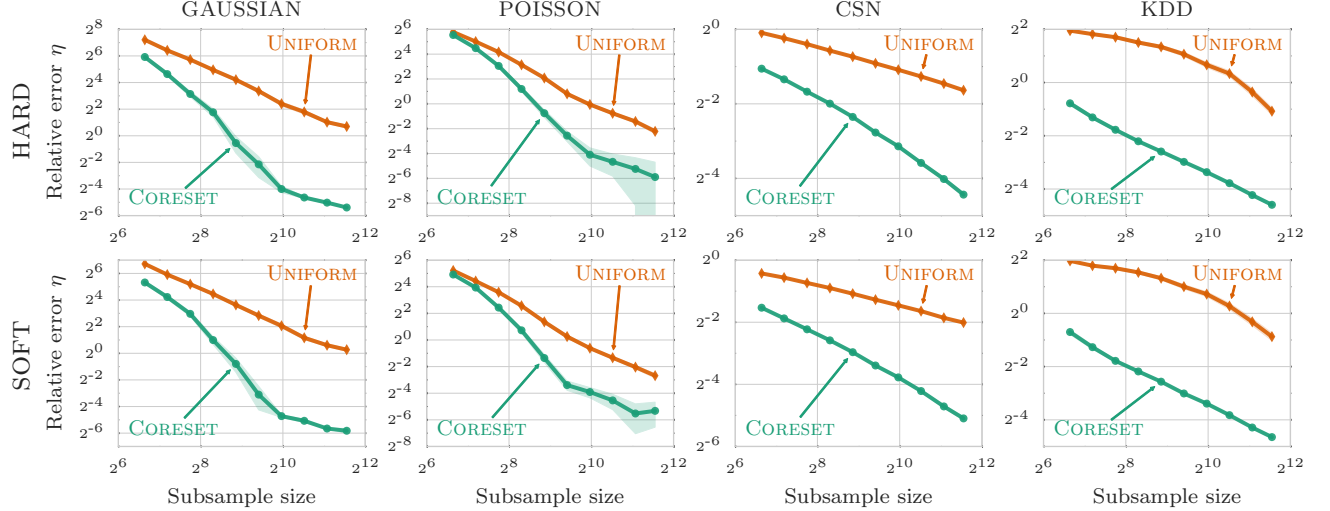


Figure 1: Relative error in relation to subsample size. For a fixed subsample size, coresets strongly outperform uniform subsampling. Shaded areas denote confidence intervals based on 500 independent trials.

5 EXPERIMENTAL EVALUATION

In this section, we demonstrate that our proposed coreset construction can be used in practice to speed up both hard and soft clustering problems. We compare our proposed coreset construction⁴ to solving the clustering problem on both uniform subsamples of the data set and the original data set itself.

We first generate a weighted subsample of the data using either uniform subsampling or the proposed coreset construction. To solve the corresponding hard (soft) clustering problem on the subsample we apply Algorithm 1 (5) with adaptive seeding using D^2 -sampling. We then evaluate the computed solution on the full data set to obtain the cost C_{ss} and measure the CPU time elapsed for both the subsampling and the solving step. We average the results over $r = 500$ independent trials. Independently, we also measure the CPU time elapsed and the solution quality C_{full} obtained when training using the full data set, again averaged across $r = 500$ independent trials. Finally, we calculate the *relative error* $\eta = (C_{ss} - C_{full})/C_{full}$ for both uniform subsampling and coresets.

5.1 Data sets and parameters

Following Banerjee et al. (2005) we use (regular) exponential family mixture models to generate two synthetic data sets. We sample the model parameters from the associated conjugate prior and cluster the data using the corresponding dual Bregman divergence (as detailed in Section 4.2).

⁴For the proposed coreset construction, we use $A = I$ for all data sets except for KDD where we use the inverse of the covariance matrix.

GAUSSIAN. This data set consists of 10,000 points which are drawn from a mixture of $k = 50$ isotropic Gaussians in $d = 10$ dimensions. We use a k -dimensional Dirichlet distribution with concentration parameter $\alpha = 0.5$ to sample the mixture weights. The means of each of the k components are in turn sampled from a Gaussian with zero mean and variance of 5000. We solve both hard and soft clustering problems with the squared Euclidean distance as the Bregman divergence and $k = 50$ cluster centers.

POISSON. This data set consists of 10,000 points drawn from a mixture of $k = 50$ multivariate Poisson distributions in $d = 10$ dimensions. In a given component j , each dimension is independently sampled, i.e. $x_i \sim \text{Poi}(\mu_{i,j})$ for each $i = 1, 2, \dots, d$. For each component j and dimension i , the parameter $\mu_{i,j}$ is sampled from a Gamma distribution with shape $\alpha = 10$ and rate $\beta = 10^{-3}$. We consider $k = 50$ cluster centers and use the relative entropy as the Bregman divergence.

CSN. In the Community Seismic Network (Faulkner et al., 2011) more than 7GB of cellphone accelerometer data were gathered and used to detect earthquakes. The data consists of 80,000 observations and 17 extracted features. We consider $k = 50$ cluster centers and use the squared Mahalanobis distance where A is the inverse of the covariance matrix.

KDD. This data set was used in the Protein Homology Prediction KDD competition and contains 145,751 training examples with 74 features that measure the match between a protein and a native sequence. We consider $k = 50$ cluster centers and use the squared Euclidean distance as the Bregman Divergence.

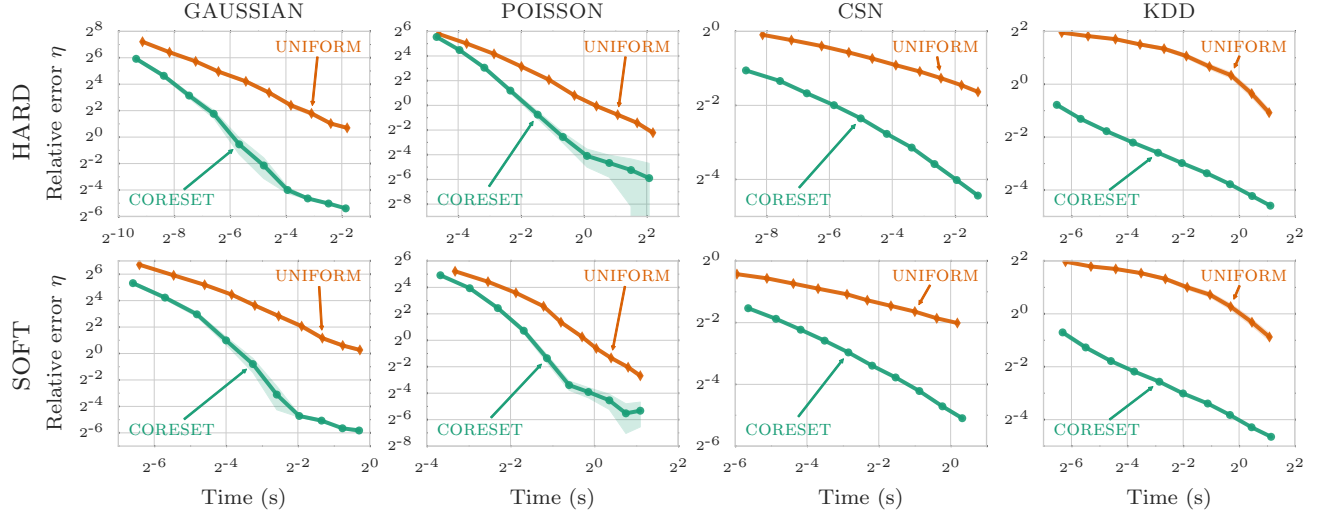


Figure 2: Time required to reach a fixed relative error using coresets and uniform subsampling. In all settings, coresets outperform uniform subsampling. Shaded areas denote confidence intervals based on 500 trials.

5.2 Discussion

Figure 1 shows the relative error η for different subsample sizes. For all data sets and both hard and soft clustering, the relative error decreases as we increase the number of points in the subsample and even for relatively small subsample sizes our coreset construction produces competitive solution compared to solving on the full data set. Across all data sets, the proposed coreset construction outperforms uniform subsampling and achieves a lower relative cost for a given subsample size. Figure 2 shows that coresets reach a given

Table 2: Hard clustering on KDD (3000 subsamples)

	UNIFORM	CORESET	FULL
TIME (s)	2.11	2.17	176.84
SPEEDUP	83.9x	81.3x	1.0x
COST (10^9)	290.46	205.18	197.02
RELATIVE ERROR η	47.4%	4.1%	0.0%

relative error faster than uniform subsamples (even if we account for the time required to construct the coreset). The practical relevance can be seen in Table 2. For KDD and a subsample size of $s = 3000$, we obtain a speedup of $81.3\times$ using coresets, while only incurring a relative error of 4.1%. At the same time, uniform subsampling leads to a relative error of 47.4%.

6 OTHER RELATED WORK

Relatively few approaches were suggested for scalable clustering with Bregman divergences. Ackermann and Blömer (2009) construct weak coresets for approximate clustering with μ -similar Bregman divergences and obtain weak coresets of size $\mathcal{O}(\frac{1}{\varepsilon^2} k \log(n) \log(k|\Gamma|^k \log n))$ where $|\Gamma| \leq n^{d \cdot \text{poly}(k/\varepsilon)}$.

For weak coresets, the approximation guarantee holds only for queries close to the optimal solution. As such, their applicability is severely limited. In contrast, we provide strong coresets for all μ -similar Bregman divergences and generalize the results to the soft clustering case. Feldman et al. (2013) provide a coreset construction (albeit, the coreset is a set of clustering features, rather than a weighted subset of the original data set) for hard clustering with μ -similar Bregman divergences with the additional restriction on the convex set S (domain of ϕ): every pair p, q of points from P, S must contain all points within a ball of radius $(4/m\varepsilon)d(p, q)$ around p for a constant μ . In contrast, our approach makes no such assumptions and provides a strong coreset with improved dependency on ε . Feldman et al. (2011) provide a coreset construction for the specific case of a mixture of semi-spherical Gaussians (bounded eigenvalues) and obtain a coreset of the size independent of the data set size.

7 CONCLUSION

We propose a single coreset construction algorithm for both hard and soft clustering with any μ -similar Bregman divergence. As a separate result, we establish the combinatorial complexity of mixture models induced by the regular exponential family. Experimental results demonstrate the practical utility of the proposed approach. In particular, the coresets outperform uniform subsampling and enjoy speedups of several orders of magnitude compared to solving on the full data set while retaining guarantees on the approximation error.

Acknowledgements

This research was partially supported by ERC StG 307036 and the Zurich Information Security Center.

References

- Ackermann, M. R. and J. Blömer (2009). Coresets and approximate clustering for Bregman divergences. In *Symposium on Discrete Algorithms (SODA)*, pp. 1088–1097. SIAM.
- Anthony, M. and P. L. Bartlett (2009). *Neural Network Learning: Theoretical Foundations*. Cambridge University Press.
- Arthur, D. and S. Vassilvitskii (2007). k -means++: The advantages of careful seeding. In *Symposium on Discrete Algorithms (SODA)*, pp. 1027–1035. SIAM.
- Bachem, O., M. Lucic, S. H. Hassani, and A. Krause (2016, February). Approximate k -means++ in sub-linear time. In *Conference on Artificial Intelligence (AAAI)*.
- Bachem, O., M. Lucic, and A. Krause (2015). Coresets for nonparametric estimation - the case of DP-means. In *International Conference on Machine Learning (ICML)*.
- Balcan, M.-F., S. Ehrlich, and Y. Liang (2013). Distributed k -means and k -median clustering on general topologies. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1995–2003.
- Banerjee, A., S. Merugu, I. S. Dhillon, and J. Ghosh (2005). Clustering with Bregman divergences. *Journal of Machine Learning Research (JMLR)* 6, 1705–1749.
- Bentley, J. L. and J. B. Saxe (1980). Decomposable searching problems I. Static-to-dynamic transformation. *Journal of Algorithms* 1(4), 301–358.
- Faulkner, M., M. Olson, R. Chandy, J. Krause, K. M. Chandy, and A. Krause (2011). The next big one: Detecting earthquakes and other rare events from community-based sensors. In *ACM/IEEE International Conference on Information Processing in Sensor Networks*.
- Feldman, D., M. Faulkner, and A. Krause (2011). Scalable training of mixture models via coresets. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 2142–2150.
- Feldman, D. and M. Langberg (2011). A unified framework for approximating and clustering data. In *Symposium on Theory of Computing (STOC)*, pp. 569–578. ACM.
- Feldman, D., M. Schmidt, and C. Sohler (2013). Turning big data into tiny data: Constant-size coresets for k -means, PCA and projective clustering. In *Symposium on Discrete Algorithms (SODA)*, pp. 1434–1453. SIAM.
- Har-Peled, S. and S. Mazumdar (2004). On coresets for k -means and k -median clustering. In *Symposium on Theory of Computing (STOC)*, pp. 291–300. ACM.
- Langberg, M. and L. J. Schulman (2010). Universal ε -approximators for integrals. In *Symposium on Discrete Algorithms (SODA)*, pp. 598–607. SIAM.
- Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28(2), 129–137.
- Lucic, M., M. I. Ohannessian, A. Karbasi, and A. Krause (2015). Tradeoffs for space, time, data and risk in unsupervised learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 663–671.
- Nielsen, F., J.-D. Boissonnat, and R. Nock (2007). On Bregman Voronoi diagrams. In *Symposium on Discrete Algorithms (SODA)*, pp. 746–755. SIAM.
- Reddi, S. J., B. Póczos, and A. Smola (2015). Communication efficient coresets for empirical loss minimization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.
- Schmitt, M. (2002). On the complexity of computing and learning with multiplicative neural networks. *Neural Computation* 14(2), 241–301.
- Wu, X., V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems* 14(1), 1–37.