Scalable Training of Mixture Models via Coresets

Dan Feldman
MIT

Matthew Faulkner
Caltech

Andreas Krause
ETH Zurich

Abstract

How can we train a statistical mixture model on a massive data set? In this paper, we show how to construct *coresets* for mixtures of Gaussians and natural generalizations. A coreset is a weighted subset of the data, which guarantees that models fitting the coreset will also provide a good fit for the original data set. We show that, perhaps surprisingly, Gaussian mixtures admit coresets of size *independent* of the size of the data set. More precisely, we prove that a weighted set of $O(dk^3/\varepsilon^2)$ data points suffices for computing a $(1+\varepsilon)$-approximation for the optimal model on the original n data points. Moreover, such coresets can be efficiently constructed in a map-reduce style computation, as well as in a streaming setting. Our results rely on a novel reduction of statistical estimation to problems in computational geometry, as well as new complexity results about mixtures of Gaussians. We empirically evaluate our algorithms on several real data sets, including a density estimation problem in the context of earthquake detection using accelerometers in mobile phones.

1 Introduction

We consider the problem of training statistical mixture models, in particular mixtures of Gaussians and some natural generalizations, on massive data sets. Such data sets may be distributed across a cluster, or arrive in a data stream, and have to be processed with limited memory. In contrast to parameter estimation for models with compact sufficient statistics, mixture models generally require inference over latent variables, which in turn depends on the full data set. In this paper, we show that Gaussian mixture models (GMMs), and some generalizations, admit small *coresets*: A coreset is a weighted subset of the data which guarantees that models fitting the coreset will also provide a good fit for the original data set. Perhaps surprisingly, we show that Gaussian mixtures admit coresets of size *independent* of the size of the data set.

We focus on ε-semi-spherical Gaussians, where the covariance matrix Σ_i of each component i has eigenvalues bounded in $[\varepsilon, 1/\varepsilon]$, but some of our results generalize even to the semi-definite case. In particular, we show that given a data set D of n points in $\mathbb{R}^d$, $\varepsilon > 0$ and $k \in \mathbb{N}$, how one can efficiently construct a weighted set C of $\mathcal{O}(dk^3/\varepsilon^2)$ points, such that for any mixture of k ε-semi-spherical Gaussians $\theta = [(w_1, \mu_1, \Sigma_1), \ldots, (w_k, \mu_k, \Sigma_k)]$ it holds that the log-likelihood $\ln P(D \mid \theta)$ of D under θ is approximated by the (properly weighted) log-likelihood $\ln P(C \mid \theta)$ of C under θ to arbitrary accuracy as $\varepsilon \to 0$. Thus solving the estimation problem on the coreset C (e.g., using weighted variants of the EM algorithm, see Section 3.3) is almost as good as solving the estimation problem on large data set D. Our algorithm for constructing C is based on adaptively sampling points from D and is simple to implement. Moreover, coresets can be efficiently constructed in a map-reduce style computation, as well as in a streaming setting (using space and update time per point of $\mathrm{poly}(dk\varepsilon^{-1} \log n \log(1/\delta))$).

Existence and construction of coresets have been investigated for a number of problems in computational geometry (such as k-means and k-median) in many recent papers (*cf.*, surveys in [1, 2]). In this paper, we demonstrate how these techniques from computational geometry can be lifted to the realm of statistical estimation. As a by-product of our analysis, we also close an open question on the VC dimension of arbitrary mixtures of Gaussians. We evaluate our algorithms on several synthetic and real data sets. In particular, we use our approach for density estimation for acceleration data, motivated by an application in earthquake detection using mobile phones.

2 Background and Problem Statement

Fitting mixture models by MLE. Suppose we are given a data set $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subseteq \mathbb{R}^d$. We consider fitting a mixture of Gaussians $\theta = [(w_1, \mu_1, \Sigma_1), \ldots, (w_k, \mu_k, \Sigma_k)]$, i.e., the distribution $P(\mathbf{x} \mid \theta) = \sum_{i=1}^k w_i \mathcal{N}(\mathbf{x}; \mu_i, \Sigma_i)$, where $w_1, \ldots, w_k \geq 0$ are the mixture weights, $\sum_i w_i = 1$, and μ_i and Σ_i are mean and covariance of the i-th mixture component, which is modeled as a multivariate normal distribution $\mathcal{N}(\mathbf{x}, \mu_i, \Sigma_i) = \frac{1}{\sqrt{|2\pi\Sigma_i|}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i)\right)$. In Section 4, we will discuss extensions to more general mixture models. Assuming the data was generated i.i.d., the negative log likelihood of the data is $\mathcal{L}(D \mid \theta) = -\sum_j \ln P(\mathbf{x}_j \mid \theta)$, and we wish to obtain the maximum likelihood estimate (MLE) of the parameters $\theta^* = \operatorname{argmin}_{\theta \in \mathfrak{C}} \mathcal{L}(D \mid \theta)$, where $\mathfrak{C}$ is a set of constraints ensuring that degenerate solutions are avoided[1]. Hereby, for a symmetric matrix $\mathbf{A}$, $\operatorname{spec} \mathbf{A}$ is the set of all eigenvalues of $\mathbf{A}$. We define

$$\mathfrak{C} = \mathfrak{C}_\varepsilon = \{\theta = [(w_1, \mu_1, \Sigma_1), \ldots, (w_k, \mu_k, \Sigma_k)] \mid \forall_i : \operatorname{spec}(\Sigma_i) \subseteq [\varepsilon, 1/\varepsilon]\}$$

to be the set of all mixtures of k Gaussians θ, such that all the eigenvalues of the covariance matrices of θ are bounded between ε and $1/\varepsilon$ for some small $\varepsilon > 0$.

Approximating the log-likelihood. Our goal is to approximate the data set D by a weighted set $C = \{(\gamma_1, \mathbf{x}'_1), \ldots, (\gamma_m, \mathbf{x}'_m)\} \subseteq \mathbb{R} \times \mathbb{R}^d$, such that $\mathcal{L}(D \mid \theta) \approx \mathcal{L}(C \mid \theta)$ for all θ, where we define $\mathcal{L}(C \mid \theta) = -\sum_i \gamma_i \ln P(\mathbf{x}'_i \mid \theta)$.

What kind of approximation accuracy may we hope to expect? Notice that there is a nontrivial issue of scale: Suppose we have a MLE θ^* for D, and let $\alpha > 0$. Then straightforward linear algebra shows that we can obtain an MLE θ^*_α for a scaled data set $\alpha D = \{\alpha \mathbf{x} : \mathbf{x} \in D\}$ by simply scaling all means by α, and covariance matrices by α^2. For the log-likelihood, however, it holds that $\mathcal{L}(\alpha D \mid \theta^*_\alpha) = d \ln \alpha + \mathcal{L}(D \mid \theta^*)$. Therefore, optimal solutions on one scale can be efficiently transformed to optimal solutions at a different scale, while maintaining the same *additive error*. This means, that any algorithm which achieves absolute error ε at any scale could be used to achieve parameter estimates (for means, covariances) with *arbitrarily* small error, simply by applying the algorithm to a scaled data set and transforming back the obtained solution. An alternative, scale-invariant approach may be to strive towards approximating $\mathcal{L}(D \mid \theta)$ up to *multiplicative error* $(1+\varepsilon)$. Unfortunately, this goal is also hard to achieve: Choosing a scaling parameter α such that $d \ln \alpha + \mathcal{L}(D \mid \theta^*) = 0$ would require any algorithm that achieves any bounded multiplicative error to essentially incur *no error at all* when evaluating $\mathcal{L}(\alpha D \mid \theta^*)$. The above observations hold even for the case $k = 1$ and $\Sigma = I$, where the mixture θ consists of a single Gaussian, and the log-likelihood is the sum of squared distances to a point μ and an additive term.

Motivated by the scaling issues discussed above, we use the following error bound that was suggested in [3] (who studied the case where all Gaussians are identical spheres). We decompose the negative log-likelihood $\mathcal{L}(D \mid \theta)$ of a data set D as

$$\mathcal{L}(D \mid \theta) = -\sum_{j=1}^n \ln \sum_{i=1}^k \frac{w_i}{\sqrt{|2\pi\Sigma_i|}} \exp\left(-\frac{1}{2}(\mathbf{x}_j - \mu_i)^T \Sigma_i^{-1} (\mathbf{x}_j - \mu_i)\right) = -n \ln Z(\theta) + \phi(D \mid \theta)$$

where $Z(\theta) = \sum_i \frac{w_i}{\sqrt{|2\pi\Sigma_i|}}$ is a normalizer, and the function ϕ is defined as

$$\phi(D \mid \theta) = -\sum_{j=1}^n \ln \sum_{i=1}^k \frac{w_i}{Z(\theta)\sqrt{|2\pi\Sigma_i|}} \exp\left(-\frac{1}{2}(\mathbf{x}_j - \mu_i)^T \Sigma_i^{-1} (\mathbf{x}_j - \mu_i)\right).$$

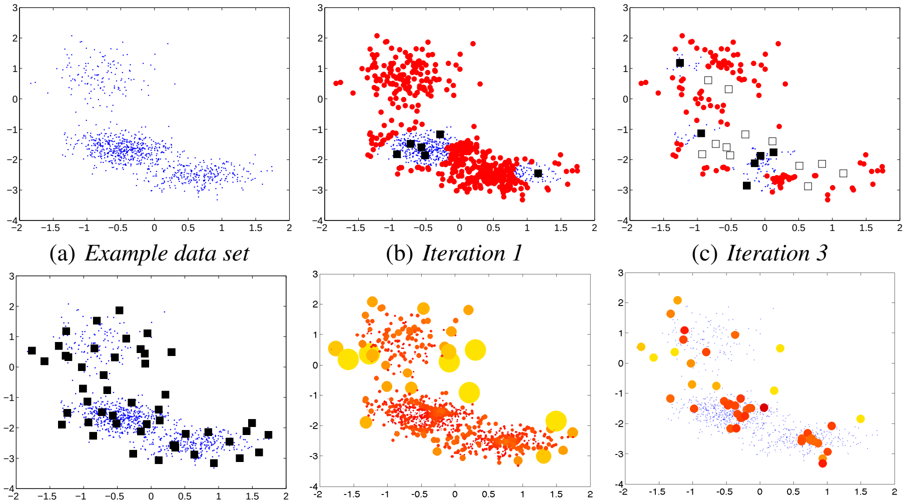
Hereby, $Z(\theta)$ plays the role of a normalizer, which can be computed *exactly*, independently of the set D. $\phi(D \mid \theta)$ captures all dependencies of $\mathcal{L}(D \mid \theta)$ on D, and via Jensen's inequality, it can be seen that $\phi(D \mid \theta)$ is always nonnegative.

We can now use this term $\phi(D \mid \theta)$ as a reference for our error bounds. In particular, we call $\tilde{\theta}$ a $(1+\varepsilon)$-approximation for θ if $(1-\varepsilon)\phi(D \mid \theta) \leq \phi(D \mid \tilde{\theta}) \leq \phi(D \mid \theta)(1+\varepsilon)$.

Coresets. We call a weighted data set C a (k, ε)-*coreset* for another (possibly weighted) set $D \subseteq \mathbb{R}^d$, if for all mixtures $\theta \in \mathfrak{C}$ of k Gaussians it holds that

$$(1-\varepsilon)\phi(D \mid \theta) \leq \phi(C \mid \theta) \leq \phi(D \mid \theta)(1+\varepsilon).$$

[1] equivalently, $\mathfrak{C}$ can be interpreted as prior thresholding.



(a) *Example data set* (b) *Iteration 1* (c) *Iteration 3*

(d) *Final approximation B* (e) *Sampling distribution* (f) *Coreset*

Figure 1: Illustration of the coreset construction for example data set (a). (b,c) show two iterations of constructing the set B. Solid squares are points sampled uniformly from remaining points, hollow squares are points selected in previous iterations. Red color indicates half the points furthest away from B, which are kept for next iteration. (d) final approximate clustering B on top of original data set. (e) Induced non-uniform sampling distribution: radius of circles indicates probability; color indicates weight, ranging from red (high weight) to yellow (low weight). (f) Coreset sampled from distribution in (e).

Hereby $\phi(C \mid \theta)$ is generalized to weighted data sets C in the natural way (weighing the contribution of each summand $\mathbf{x}'_j \in C$ by γ_j). Thus, as $\varepsilon \to 0$, for a sequence of (k, ε)-coresets C_ε we have that $\sup_{\theta \in \mathfrak{C}} |\mathcal{L}(C_\varepsilon \mid \theta) - \mathcal{L}(D \mid \theta)| \to 0$, i.e., $\mathcal{L}(C_\varepsilon \mid \theta)$ uniformly (over $\theta \in \mathfrak{C}$) approximates $\mathcal{L}(D \mid \theta)$. Further, under the additional condition that all variances are sufficiently large (formally $\prod_{\lambda \in \mathrm{spec}(\Sigma_i)} \lambda \geq \frac{1}{(2\pi)^d}$ for all components i), the log-normalizer $\ln Z(\theta)$ is negative, and consequently the coreset in fact provides a multiplicative $(1+\varepsilon)$ approximation to the log-likelihood, i.e.,

$$(1-\varepsilon)\mathcal{L}(D \mid \theta) \leq \mathcal{L}(C \mid \theta) \leq \mathcal{L}(D \mid \theta)(1+\varepsilon).$$

More details can be found in the Appendix (see Observation C.6).

Note that if we had access to a (k, ε)-coreset C, then we could reduce the problem of fitting a mixture model on D to one of fitting a model on C, since the optimal solution θ_C is a good approximation (in terms of log-likelihood) of θ^*. While finding the optimal θ_C is a difficult problem, one can use a (weighted) variant of the EM algorithm to find a good solution. Moreover, if $|C| \ll |D|$, running EM on C may be orders of magnitude faster than solving it on D. In Section 3.3, we give more details about solving the density estimation problem on the coreset.

The key question is whether small (k, ε)-coresets exist, and whether they can be efficiently constructed. In the following, we answer this question affirmatively. We show that, perhaps surprisingly, one can efficiently find coresets C of size *independent* of the size n of D, and with polynomial dependence on $\frac{1}{\varepsilon}$, d and k.

3 Efficient Coreset Construction via Adaptive Sampling

Naive approach: uniform sampling. A naive approach towards approximating D would be to just pick a subset C uniformly at random. In particular, suppose the data set is generated from a mixture of two spherical Gaussians ($\Sigma_i = \mathbf{I}$) with weights $w_1 = \frac{1}{\sqrt{n}}$ and $w_2 = 1 - \frac{1}{\sqrt{n}}$. Unless $m = \Omega(\sqrt{n})$ points are sampled, with constant probability no data point generated from Gaussian 2 is selected. By moving the means of the Gaussians arbitrarily far apart, $\mathcal{L}(D \mid \theta_C)$ can be made arbitrarily worse than $\mathcal{L}(D \mid \theta_D)$, where θ_C and θ_D are MLEs on C and D respectively. Thus, even for two well-separated Gaussians, uniform sampling can perform arbitrarily poorly. This example already suggests that, intuitively, in order to achieve small multiplicative error, we must devise a sampling scheme that adaptively selects representative points from all "clusters" present in the data set. However, this suggests that obtaining a coreset requires solving a chicken-and-egg problem, where we need to understand the density of the data to obtain the coreset, but simultaneously would like to use the coreset for density estimation.

Better approximation via adaptive sampling. The key idea behind the coreset construction is that we can break the chicken-and-egg problem by first obtaining a rough approximation B of the clustering solution (using more than k components, but far fewer than n), and then to use this solution to bias the random sampling. Surprisingly, a simple procedure which iteratively samples a small number β of points, and removes half of the data set closest to the sampled points, provides a sufficiently accurate first approximation B for this purpose. This initial clustering is then used to sample the data points comprising coreset C according to probabilities which are roughly proportional to the squared distance to the set B. This non-uniform random sampling can be understood as an importance-weighted estimate of the log-likelihood $\mathcal{L}(D \mid \theta)$, where the weights are optimized in order to reduce the variance. The same general idea has been found successful in constructing coresets for geometric clustering problems such as k-means and k-median [4]. The pseudocode for obtaining the approximation B, and for using it to obtain coreset C is given in Algorithm 1.

Algorithm 1: Coreset construction

```
Input: Data set $D$, $\varepsilon$, $\delta$, $k$
Output: Coreset $C = \{(\gamma(\mathbf{x}_1), \mathbf{x}_1), \dots, (\gamma(\mathbf{x}_{|C|}), \mathbf{x}_{|C|})\}$
$D' \leftarrow D$; $B \leftarrow \emptyset$;
while $|D'| > 10dk \ln(1/\delta)$ do
    Sample set $S$ of $\beta = 10dk \ln(1/\delta)$ points uniformly at random from $D'$;
    Remove $\lceil |D'|/2 \rceil$ points $\mathbf{x} \in D'$ closest to $S$ (i.e., minimizing $\mathrm{dist}(\mathbf{x}, S)$) from $D'$;
    Set $B \leftarrow B \cup S$;
Set $B \leftarrow B \cup D'$;
for each $b \in B$ do  $D_b \leftarrow$ the points in $D$ whose closest point in $B$ is $b$. Ties broken arbitrarily;
for each $b \in B$ and $\mathbf{x} \in D_b$ do
    $m(\mathbf{x}) \leftarrow \left\lceil \frac{5}{|D_b|} + \frac{\mathrm{dist}(\mathbf{x},B)^2}{\sum_{\mathbf{x}' \in D} \mathrm{dist}(\mathbf{x}',B)^2} \right\rceil$;
Pick a non-uniform random sample $C$ of $10\lceil dk|B|^2 \ln(1/\delta)/\varepsilon^2 \rceil$ points from $D$, where for every $\mathbf{x}' \in C$ and
$\mathbf{x} \in D$, we have $\mathbf{x}' = \mathbf{x}$ with probability $m(\mathbf{x}) / \sum_{\mathbf{x}' \in D} m(\mathbf{x}')$;
for each $\mathbf{x}' \in C$ do  $\gamma(\mathbf{x}') \leftarrow \frac{\sum_{\mathbf{x} \in D} m(\mathbf{x})}{|C| \cdot m(\mathbf{x}')}$;
```

We have the following result, proved in the Appendix:

Theorem 3.1. *Suppose C is sampled from D using Algorithm 1 for parameters ε, δ and k. Then, with probability at least $1 - \delta$ it holds that for all $\theta \in \mathfrak{C}_\varepsilon$,*

$$\phi(D \mid \theta)(1 - \varepsilon) \leq \phi(C \mid \theta) \leq \phi(D \mid \theta)(1 + \varepsilon).$$

In our experiments, we compare the performance of clustering on coresets constructed via adaptive sampling, vs. clustering on a uniform sample. The size of C in Algorithm 1 depends on $|B|^2 = \log^2 n$. By replacing B in the algorithm with a constant factor approximation B', $|B'| = l$ for the k-means problem, we can get a coreset C of size *independent* of n. Such a set B' can be computed in $O(ndk)$ time either by applying exhaustive search on the output C of the original Algorithm 1 or by using one of the existing constant-factor approximation algorithms for k-means (say, [5]).

3.1 Sketch of Analysis: Reduction to Euclidean Spaces

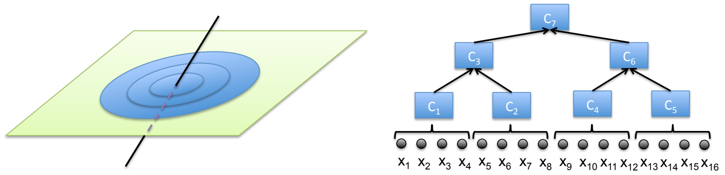
For space limitations, the proof of Theorem 3.1 is included in the supplemental material, we only provide a sketch of the analysis, carrying the main intuition. The key insight in the proof is that the contribution $\log P(\mathbf{x} \mid \theta)$ to the likelihood $\mathcal{L}(D \mid \theta)$ can be expressed in the following way:

Lemma 3.2. *There exist functions ϕ, ψ, and f such that, for any point $\mathbf{x} \in \mathbb{R}^d$ and mixture model θ, $\ln P(\mathbf{x} \mid \theta) = -f_{\phi(\mathbf{x})}(\psi(\theta)) + Z(\theta)$, where*

$$f_{\tilde{\mathbf{x}}}(y) = -\ln \sum_i \tilde{w}_i \exp\left(-W_i \mathrm{dist}(\tilde{\mathbf{x}} - \tilde{\mu}_i, \mathbf{s}_i)^2\right).$$

Hereby, ϕ is a function that maps a point $\mathbf{x} \in \mathbb{R}^d$ into $\tilde{\mathbf{x}} = \phi(\mathbf{x}) \in \mathbb{R}^{2d}$, and ψ is a function that maps a mixture model θ into a tuple $y = (s, w, \tilde{\mu}, W)$ where w is a k-tuple of nonnegative weights $\tilde{w}_1, \dots, \tilde{w}_k$ summing to 1, $s = \mathbf{s}_1, \dots, \mathbf{s}_k \subseteq \mathbb{R}^{2d}$ is a set of k d-dimensional subspaces that are weighted by weights $W_1, \cdots, W_k > 0$, and $\tilde{\mu} = \tilde{\mu}_1, \cdots, \tilde{\mu}_k \in \mathbb{R}^{2d}$ is a set of k means.

The main idea behind Lemma 3.2 is that level sets of distances between points and subspaces are quadratic forms, and can thus represent level sets of the Gaussian probability density function (see Figure 2(a) for an illustration). We recognize the "soft-min" function $\wedge_{\mathbf{w}'}(\eta) \equiv$



(a) *Gaussian pdf as Euclidean distances* (b) *Tree for coreset construction*

Figure 2: (a) Level sets of the distances between points on a plane (green) and (disjoint) k-dimensional subspaces are ellipses, and thus can represent contour lines of the multivariate Gaussian. (b) Tree construction for generating coresets in parallel or from data streams. Black arrows indicate "merge-and-compress" operations. The (intermediate) coresets $C_1, \ldots, C_7$ are enumerated in the order in which they would be generated in the streaming case. In the parallel case, C_1, C_2, C_4 and C_5 would be constructed in parallel, followed by parallel construction of C_3 and C_6, finally resulting in C_7.

$-\ln \sum_i w'_i \exp(-\eta_i)$ as an approximation upper-bounding the minimum $\min(\eta) = \min_i \eta_i$ for $\eta_i = W_i \mathrm{dist}(\tilde{\mathbf{x}} - \tilde{\mu}_i, \mathbf{s}_i)^2$ and $\eta = [\eta_1, \ldots, \eta_k]$. The motivation behind this transformation is that it allows expressing the likelihood $P(\mathbf{x} \mid \theta)$ of a data point $\mathbf{x}$ given a model θ in a purely geometric manner as soft-min over distances between points and subspaces in a transformed space. Notice that if we use the minimum $\min()$ instead of the soft-min $\wedge_{\tilde{w}}()$, we recover the problem of approximating the data set D (transformed via ϕ) by k-subspaces. For semi-spherical Gaussians, it can be shown that the subspaces can be chosen as points while incurring a multiplicative error of at most $1/\varepsilon$, and thus we recover the well-known k-means problem in the transformed space. This insight suggests using a known coreset construction for k-means, adapted to the transformation employed. The remaining challenge in the proof is to bound the additional error incurred by using the soft-min function $\wedge_{\tilde{w}}(\cdot)$ instead of the minimum $\min(\cdot)$. We tackle this challenge by proving a generalized triangle inequality adapted to the exponential transformation, and employing the framework described in [4], which provides a general method for constructing coresets for clustering problems of the form $\min_{\mathbf{s}} \sum_i f_{\tilde{\mathbf{x}}}(\mathbf{s})$.

As proved in [4], the key quantity that controls the size of a coreset is the pseudo-dimension of the functions $F_d = \{f_{\tilde{\mathbf{x}}} \text{ for } \tilde{\mathbf{x}} \in \mathbb{R}^{2d}\}$. This notion of dimension is closely related to the VC dimension of the (sub-level sets of the) functions F_d and therefore represents the complexity of this set of functions. The final ingredient in the proof of Theorem 3.1 is Theorem C.4 in the supplemental material, which provides a new bound on the complexity of mixtures of k Gaussians in d dimensions.

3.2 Streaming and Parallel Computation

One major advantage of coresets is that they can be constructed in parallel, as well as in a streaming setting where data points arrive one by one, and it is impossible to remember the entire data set due to memory constraints. The key insight is that coresets satisfy certain composition properties, which have previously been used by [6] for streaming and parallel construction of coresets for geometric clustering problems such as k-median and k-means.

1. Suppose C_1 is a (k, ε)-coreset for D_1, and C_2 is a (k, ε)-coreset for D_2. Then $C_1 \cup C_2$ is a (k, ε)-coreset for $D_1 \cup D_2$.
2. Suppose C is a (k, ε)-coreset for D, and C' is a (k, δ)-coreset for C. Then C' is a $(k, (1 + \varepsilon)(1 + \delta) - 1)$-coreset for D.

In the following, we review how to exploit these properties for parallel and streaming computation.

Streaming. In the streaming setting, we assume that points arrive one-by-one, but we do not have enough memory to remember the entire data set. Thus, we wish to maintain a coreset over time, while keeping only a small subset of $\mathcal{O}(\log n)$ coresets in memory. There is a general reduction that shows that a small coreset scheme to a given problem suffices to solve the corresponding problem on a streaming input [7, 6]. The idea is to construct and save in memory a coreset for every block of $\mathrm{poly}(dk/\varepsilon)$ consecutive points arriving in a stream. When we have two coresets in memory, we can merge them (resulting in a (k, ε)-coreset via property (1)), and compress by computing a single coreset from the merged coresets (via property (2)) to avoid increase in the coreset size. An important subtlety arises: While merging two coresets (via property (1)) does not increase the approximation error, compressing a coreset (via property (2)) *does* increase the error. A naive approach that merges and compresses immediately as soon as two coresets have been constructed, can incur an exponential increase in approximation error. Fortunately, it is possible to organize the merge-and-compress operations in a binary tree of height $O(\log n)$, where we need to store in memory a single coreset for each level on the tree (thus requiring only $\mathrm{poly}(dk\varepsilon^{-1}\log n)$ memory). Figure 2(b) illustrates this tree computation. In order to construct a coreset for the union of two (weighted) coresets, we use a weighted version of Algorithm 1, where we consider a weighted point as duplicate copies of a non-weighted point (possibly with fractional weight). A more formal description can be found in [8]. We summarize our streaming result in the following theorem.

Theorem 3.3. *A (k, ε)-coreset for a stream of n points in $\mathbb{R}^d$ can be computed for the ε-semi-spherical GMM problem with probability at least $1 - \delta$ using space and update time* $\mathrm{poly}(dk\varepsilon^{-1}\log n\log(1/\delta))$.

Parallel/Distributed computations. Using the same ideas from the streaming model, a (non-parallel) coreset construction can be transformed into a parallel one. We partition the data into sets, and compute coresets for each set, independently, on different computers in a cluster. We then (in parallel) merge (via property (1)) two coresets, and compute a single coreset for every pair of such coresets (via property (2)). Continuing in this manner yields a process that takes $\mathcal{O}(\log n)$ iterations of parallel computation. This computation is also naturally suited for map-reduce [9] style computations, where the map tasks compute coresets for disjoint parts of D, and the reduce tasks perform the merge-and-compress operations. Figure 2(b) illustrates this parallel construction.

Theorem 3.4. *A (k, ε)-coreset for a set of n points in $\mathbb{R}^d$ can be computed for the ε-semi-spherical GMM problem with probability at least $1 - \delta$ using m machines in time $(n/m) \cdot$* $\mathrm{poly}(dk\varepsilon^{-1}\log(1/\delta)\log n)$.

3.3 Fitting a GMM on the Coreset using Weighted EM

One approach, which we employ in our experiments, is to use a natural generalization of the EM algorithm, which takes the coreset weights into account. We here describe the algorithm for the case of GMMs. For other mixture distributions, the E and M steps are modified appropriately.

Algorithm 2: Weighted EM for Gaussian mixtures

```
Input: Coreset $C$, $k$, TOL
Output: Mixture model $\theta_C$
$\mathcal{L}_{old} = \infty$; Initialize means $\mu_1, \ldots, \mu_k$ by sampling $k$ points from $C$ with probability proportional to their weight. Initialize $\Sigma_i = I$ and $w_i = \frac{1}{k}$ for all $i$;
repeat
    $\mathcal{L}_{old} = \mathcal{L}(C \mid \theta)$; for $j = 1$ to $n$ do  for $i = 1$ to $k$ do  Compute $\eta_{i,j} = \gamma_i \frac{w_i \mathcal{N}(\mathbf{x}'_j;\mu_i,\Sigma_i)}{\sum_\ell w_\ell \mathcal{N}(\mathbf{x}'_j;\mu_\ell,\Sigma_\ell)}$;
    for $i = 1$ to $k$ do
        $w_i \leftarrow w_i / \sum_\ell w_i$; $\mu_i \leftarrow \sum_j \eta_{i,j}\mathbf{x}'_j / \sum_j \eta_{i,j}$; $\Sigma_i \leftarrow \sum_j \eta_{i,j}\left(\mathbf{x}'_j - \mu_i\right)\left(\mathbf{x}'_j - \mu_i\right)^T / \sum_j \eta_{i,j}$;
until $\mathcal{L}(C \mid \theta) \geq \mathcal{L}_{old} - TOL$ ;
```

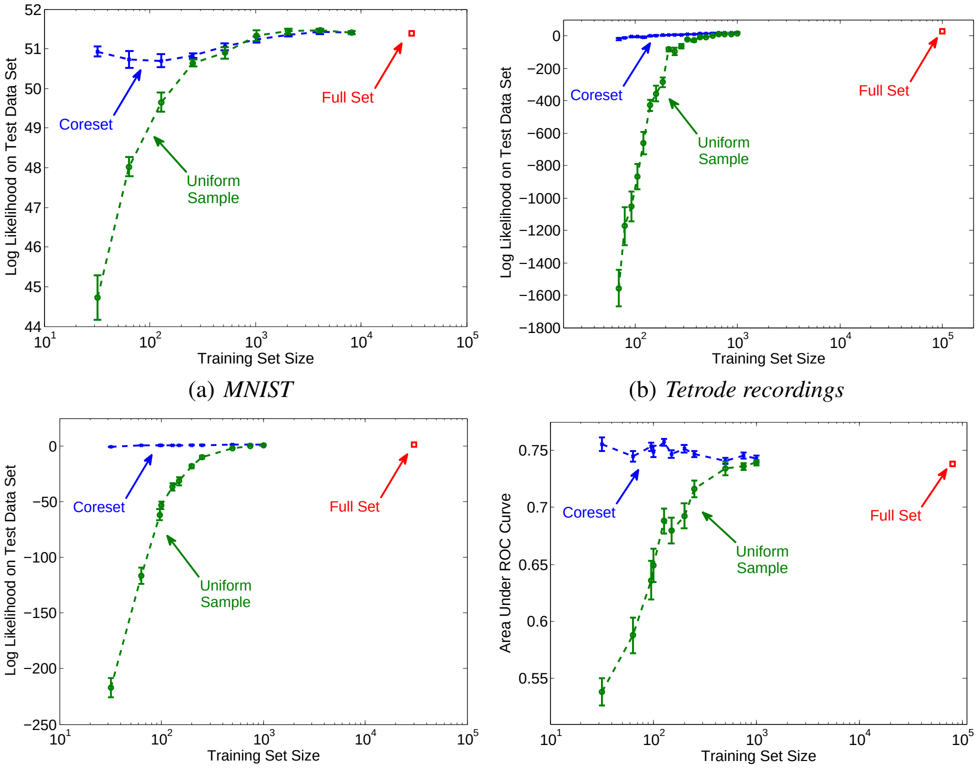
Using a similar analysis as for the standard EM algorithm, Algorithm 2 is guaranteed to converge, but only to a local optimum. However, since it is applied on a much smaller set, it can be initialized using multiple random restarts.

4 Extensions and Generalizations

We now show how the connection between estimating the parameters for mixture models and problems in computational geometry can be leveraged further. Our observations are based on the link between mixture of Gaussians and projective clustering (multiple subspace approximation) as shown in Lemma 3.2.

Generalizations to non-semi-spherical GMMs. For simplicity, we generalized the coreset construction for the k-means problem, which required assumptions that the Gaussians are ε-semi-spherical. However, several more complex coresets for projective clustering were suggested recently (*cf.*, [4]). Using the tools developed in this article, each such coreset implies a corresponding coreset for GMMs and generalizations. As an example, the coresets for approximating points by lines [10] implies that we can construct small coresets for GMMs even if the smallest singular value of one of the corresponding covariance matrices is zero.

Generalizations to ℓ_q distances and other norms. Our analysis is based on combinatorics (such as the complexity of sub-levelsets of GMMs) and probabilistic methods (non-uniform random sampling). Therefore, generalizations to other non-Euclidean distance functions, or error functions such as (non-squared) distances (mixture of Laplace distributions) is straightforward. The main property that we need is a generalization of the triangle inequality, as proved in



(a) *MNIST* (b) *Tetrode recordings*

(c) *CSN data* (d) *CSN detection*

Figure 3: Experimental results for three real data sets. We compare likelihood of the best model obtained on subsets C constructed by uniform sampling, and by the adaptive coreset sampling procedure.

Lemma A.4. For example, replacing the squared distances by non-squared distances in Lemma A.4 and its proof yields a coreset for mixture of Laplace distributions. The double triangle inequality $\|a-c\|^2 \le 2(\|a-b\| + \|b-c\|^2)$ that we used in this paper is replaced by Hölder's inequality, $\|a-c\|^2 \le 2^{O(q)} \|a-b\| + 2\|b-c\|^2$. Such a result is straight-forward from our analysis, and we summarize it in the following theorem.

Theorem 4.1. *Let $q \ge 1$ be an integer. Consider Algorithm 1, where $\mathrm{dist}(\cdot,\cdot)^2$ is replaced by $\mathrm{dist}(\cdot,\cdot)^q$ and ε^2 is replaced by $\varepsilon^{O(q)}$. Suppose C is sampled from D using this updated version of Algorithm 1 for parameters ε, δ and k. Then, with prob. at least $1-\delta$ it holds that for all $\theta \in \mathfrak{C}_\varepsilon$,*

$$\phi(D \mid \theta)(1-\varepsilon) \le \phi(C \mid \theta) \le \phi(D \mid \theta)(1+\varepsilon),$$

where $Z(\theta) = \sum_i \frac{w_i}{g(\theta_i)}$ and $\phi(D \mid \theta) = -\sum_{\mathbf{x} \in D} \ln \sum_{i=1}^{k} \frac{w_i}{Z(\theta) g(\theta_i)} \exp\left(-\frac{1}{2}\left\|\Sigma_i^{-1/2}(\mathbf{x}-\mu_i)\right\|^q\right)$ using the normalizer $g(\theta_i) = \int \exp\left(-\frac{1}{2}\left\|\Sigma_i^{-1/2}(\mathbf{x}-\mu_i)\right\|^q\right) d\mathbf{x}$.

5 Experiments

We experimentally evaluate the effectiveness of using coresets of different sizes for training mixture models. We compare against running EM on the full set, as well as on an unweighted, uniform sample from D. Results are presented for three real datasets.

MNIST handwritten digits. The MNIST dataset contains 60,000 training and 10,000 testing grayscale images of handwritten digits. As in [11], we normalize each component of the data to have zero mean and unit variance, and then reduce each 784-pixel (28x28) image using PCA, retaining only the top $d = 100$ principal components as a feature vector. From the training set, we produce coresets and uniformly sampled subsets of sizes between 30 and 5000, using the parameters $k = 10$ (a cluster for each digit), $\beta = 20$ and $\delta = 0.1$ (see Algorithm 1), and fit GMMs using EM with 3 random restarts. The log likelihood (LLH) of each model on the testing data is shown in Figure 3(a). Notice that coresets significantly outperform uniform samples of the same size, and even a coreset of 30 points performs very well. Further note how the test-log likelihood begins to flatten out for

$|C| = 1000$. Constructing the coreset and running EM on this size takes 7.9 seconds (Intel Xeon 2.6 GHz), over 100 times faster than running EM on the full set (15 minutes).

Neural tetrode recordings. We also compare coresets and uniform sampling on a large dataset containing 319,209 records of rat hippocampal action potentials, measured by four co-located electrodes. As done by [11], we concatenate the 38-sample waveforms produced by each electrode to obtain a 152-dimensional vector. The vectors are normalized so each component has zero mean and unit variance. The 319,209 records are divided in half to obtain training and testing sets. From the training set, we produce coresets and uniformly sampled subsets of sizes between 70 and 1000, using the parameters $k = 33$ (as in [11]), $\beta = 66$, and $\delta = 0.1$, and fit GMMs. The log likelihood of each model on the held-out testing data is shown in Figure 3(b). Coreset GMMs obtain consistently higher LLH than uniform sample GMMs for sets of the same size, and even a coreset of 100 points performs very well. Overall, training on coresets achieves approximately the same likelihood as training on the full set about 95 times faster (1.2 minutes vs. 1.9 hours).

CSN cell phone accelerometer data. Smart phones with accelerometers are being used by the Community Seismic Network (CSN) as inexpensive seismometers for earthquake detection. In [12], 7 GB of acceleration data were recorded from volunteers while carrying and operating their phone in normal conditions (walking, talking, on desk, etc.). From this data, 17-dimensional feature vectors were computed (containing frequency information, moments, etc.). The goal is to train, in an online fashion, GMMs based on normal data, which then can be used to perform anomaly detection to detect possible seismic activity. Motivated by the limited storage on smart phones, we evaluate coresets on a data set of 40,000 accelerometer feature vectors, using the parameters $k = 6$, $\beta = 12$, and $\delta = 0.1$. Figure 3(c) presents the results of this experiment. Notice that on this data set, coresets show an even larger improvement over uniform sampling. We hypothesize that this is due to the fact that the recorded accelerometer data is imbalanced, and contains clusters of vastly varying size, so uniform sampling does not represent smaller clusters well. Overall, the coresets obtain a speedup of approximately 35 compared to training on the full set. We also evaluate how GMMs trained on the coreset compare with the baseline GMMs in terms of anomaly detection performance. For each GMM, we compute ROC curves measuring the performance of detecting earthquake recordings from the Southern California Seismic Network (*cf.*, [12]). Note that even very small coresets lead to performance comparable to training on the full set, drastically outperforming uniform sampling (Fig. 3(d)).

6 Related Work

Theoretical results on mixtures of Gaussians. There has been a significant amount of work on learning and applying GMMs (and more general distributions). Perhaps the most commonly used technique in practice is the EM algorithm [13], which is however only guaranteed to converge to a local optimum of the likelihood. Dasgupta [14] is the first to show that parameters of an unknown GMM P can be estimated in polynomial time, with arbitrary accuracy ε, given i.i.d. samples from P. However, his algorithm assumes a common covariance, bounded excentricity, a (known) bound on the smallest component weight, as well as a separation (distance of the means), that scales as $\Omega(\sqrt{d})$. Subsequent works relax the assumption on separation to $d^{1/4}$ [15] and $k^{1/4}$ [16]. [3] is the first to learn general GMMs, with separation $d^{1/4}$. [17] provides the first result that does not require any separation, but assumes that the Gaussians are axis-aligned. Recently, [18] and [19] provide algorithms with polynomial running time (except exponential dependence on k) and sample complexity for arbitrary GMMs. However, in contrast to our results, all the results described above crucially rely on the fact that the data set D is actually generated by a mixture of Gaussians. The problem of fitting a mixture model with near-optimal log-likelihood for arbitrary data is studied by [3], who provides a PTAS for this problem. However, their result requires that the Gaussians are identical spheres, in which case the maximum likelihood problem is identical to the k-means problem. In contrast, our results make only mild assumptions about the Gaussian components. Furthermore, none of the algorithms described above applies to the streaming or parallel setting.

Coresets. Approximation algorithms in computational geometry often make use of random sampling, feature extraction, and ϵ-samples [20]. Coresets can be viewed as a general concept that includes all of the above, and more. See a comprehensive survey on this topic in [4]. It is not clear that there is any commonly agreed-upon definition of a coreset, despite several inconsistent attempts to do so [6, 8]. Coresets have been the subject of many recent papers and several surveys [1, 2]. They have been used to great effect for a host of geometric and graph problems, including k-median [6], k-mean [8], k-center [21], k-line median [10] subspace approximation [10, 22], etc. Coresets also imply streaming algorithms for many of these problems [6, 1, 23, 8]. A framework that generalizes and improves several of these results has recently appeared in [4].

7 Conclusion

We have shown how to construct coresets for estimating parameters of GMMs and natural generalizations. Our construction hinges on a natural connection between statistical estimation and clustering problems in computational geometry. To our knowledge, our results provide the first rigorous guarantees for obtaining compressed ε-approximations of the log-likelihood of mixture models for large data sets. The coreset construction relies on an intuitive adaptive sampling scheme, and can be easily implemented. By exploiting certain closure properties of coresets, it is possible to construct them in parallel, or in a single pass through a stream of data, using only $\mathrm{poly}(dk\varepsilon^{-1}\log n \log(1/\delta))$ space and update time. Unlike most of the related work, our coresets provide guarantees for any given (possibly unstructured) data, without assumptions on the distribution or model that generated it. Lastly, we apply our construction on three real data sets, demonstrating significant gains over no or naive subsampling.

Acknowledgments This research was partially supported by ONR grant N00014-09-1-1044, NSF grants CNS-0932392, IIS-0953413 and DARPA MSEE grant FA8650-11-1-7156.

References

[1] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximations via coresets. *Combinatorial and Computational Geometry - MSRI Publications*, 52:1–30, 2005.

[2] A. Czumaj and C. Sohler. Sublinear-time approximation algorithms for clustering via random sampling. *Random Struct. Algorithms (RSA)*, 30(1-2):226–256, 2007.

[3] Sanjeev Arora and Ravi Kannan. Learning mixtures of separated nonspherical gaussians. *Annals of Applied Probability*, 15(1A):69–92, 2005.

[4] D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *Proc. 41th Annu. ACM Symp. on Theory of Computing (STOC)*, 2011.

[5] S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007.

[6] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proc. 36th Annu. ACM Symp. on Theory of Computing (STOC)*, pages 291–300, 2004.

[7] Jon Louis Bentley and James B. Saxe. Decomposable searching problems i: Static-to-dynamic transformation. *J. Algorithms*, 1(4):301–358, 1980.

[8] D. Feldman, M. Monemizadeh, and C. Sohler. A PTAS for k-means clustering based on weak coresets. In *Proc. 23rd ACM Symp. on Computational Geometry (SoCG)*, pages 11–18, 2007.

[9] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. In *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, 2004.

[10] D. Feldman, A. Fiat, and M. Sharir. Coresets for weighted facilities and their applications. In *Proc. 47th IEEE Annu. Symp. on Foundations of Computer Science (FOCS)*, pages 315–324, 2006.

[11] Ryan Gomes, Andreas Krause, and Pietro Perona. Discriminative clustering by regularized information maximization. In *Proc. Neural Information Processing Systems (NIPS)*, 2010.

[12] Matthew Faulkner, Michael Olson, Rishi Chandy, Jonathan Krause, K. Mani Chandy, and Andreas Krause. The next big one: Detecting earthquakes and other rare events from community-based sensors. In *In Proc. ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2011.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Statist. Soc. Ser. B*, 39:1–38, 1977.

[14] S. Dasgupta. Learning mixtures of gaussians. In *Fortieth Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1999.

[15] S. Dasgupta and L.J. Schulman. A two-round variant of em for gaussian mixtures. In *Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.

[16] S. Vempala and G. Wang. A spectral algorithm for learning mixture models. In *In Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002.

[17] J. Feldman, R. A. Servedio, and R. O'Donnell. Pac learning axis-aligned mixtures of gaussians with no separation assumption. In *COLT*, 2006.

[18] A. Moitra and G. Valiant. Settling the polynomial learnability of mixtures of gaussians. In *In Proc. Foundations of Computer Science (FOCS)*, 2010.

[19] M. Belkin and K. Sinha. Polynomial learning of distribution families. In *In Proc. Foundations of Computer Science (FOCS)*, 2010.

[20] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.

[21] S. Har-Peled and K. R. Varadarajan. High-dimensional shape fitting in linear time. *Discrete & Computational Geometry*, 32(2):269–288, 2004.

[22] M.W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697, 2009.

[23] G. Frahling and C. Sohler. Coresets in dynamic geometric data streams. In *Proc. 37th Annu. ACM Symp. on Theory of Computing (STOC)*, pages 209–217, 2005.

[24] D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. In *Annu. ACM Symp. on Computational Geometry (SoCG)*, 1986.

[25] Yi Li, Philip M. Long, and Aravind Srinivasan. Improved bounds on the sample complexity of learning. In *Symp. on Discrete Algorithms*, pages 309–318, 2000.

[26] M. Schmitt. On the complexity of computing and learning with multiplicative neural networks. *Neural Computation*, 14(2):241–301, 2002.

A A Geometric Approach for Mixture of Gaussians

Let $\tilde{D} = \{\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_n\} \subseteq \mathbb{R}^{2d}$ and S be a set of subspaces of $\mathbb{R}^{2d}$. Let Y be the set of all 4-tuples $(s, w, \tilde{\mu}, W)$, where $s = (s_1, \ldots, s_k) \in S^k$, $\tilde{\mu} = (\tilde{\mu}_1, \cdots, \tilde{\mu}_k)$, $W = (W_1, \cdots, W_k) \in \mathbb{R}^k$, $w = (w_1, \cdots, w_k) \in \mathbb{R}^k$, $\sum_{i=1}^k w_i = 1$ and $w_i, W_i \geq 0$, and $\tilde{\mu}_i \in \mathbb{R}^{2d}$ for every i, $1 \leq i \leq k$.

For every $\tilde{\mathbf{x}} \in \mathbb{R}^{2d}$ and a set $X \subseteq \mathbb{R}^{2d}$, we define $\operatorname{dist}(\tilde{\mathbf{x}}, X) = \min_{q \in X} \operatorname{dist}(\tilde{\mathbf{x}}, q)$. For a tuple $y = (s, w, \tilde{\mu}, W) \in Y$, we define

$$\operatorname{dist}(\tilde{\mathbf{x}}, y) = \min_{1 \leq i \leq k} W_i \operatorname{dist}(\tilde{\mathbf{x}} - \tilde{\mu}_i, s_i).$$

For every $\tilde{\mathbf{x}} \in \mathbb{R}^{2d}$ and $y = (s, w, \tilde{\mu}, W) \in Y$ we define

$$f_{\tilde{\mathbf{x}}}(y) = -\ln\left(\sum_{i=1}^{k} w_i \exp\left(-W_i \operatorname{dist}(\tilde{\mathbf{x}} - \tilde{\mu}_i, s_i)^2\right)\right).$$

Note that a flat (affine subspace) can be represented by a subspace s_1 and a translation vector $\tilde{\mu}$. The distance from a point $\mathbf{x}$ to the flat is then the distance between $(\mathbf{x} - \tilde{\mu})$ to the subspace s_1.

Lemma A.1. *Let $\Sigma \in \mathbb{R}^{d \times d}$ be a covariance matrix whose smallest singular value is $W_{\min}^2 > 0$. Let $\mu \in \mathbb{R}^d$. Then there is a vector $\tilde{\mu} \in \mathbb{R}^{2d}$ and a d-dimensional subspace s of $\mathbb{R}^{2d}$ such that for every $\mathbf{x} \in \mathbb{R}^d$ and a corresponding $\tilde{\mathbf{x}} = (\mathbf{x}, 0, \ldots, 0) \in \mathbb{R}^{2d}$ we have*

$$(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu) = W_{\min}^{-1/2} \operatorname{dist}(\tilde{\mathbf{x}} - \tilde{\mu}, s)^2.$$

Proof. Let $\Sigma^{-1} = LL^T$ be the Cholesky decomposition of Σ^{-1}, $L = UDV^T$ denote the SVD of L, and $D_{i,i}$ denote the ith entry of the diagonal of D, $1 \leq i \leq d$. Let $A^T = W_{\min} D^T U^T$ and $\tilde{\mu} = (\mu, 0, \ldots, 0) \in \mathbb{R}^{2d}$. Since $\Sigma = (LL^T)^{-1} = UD^{-2}U^T$, we have

$$\frac{1}{\max\left\{D_{1,1}^2, \ldots, D_{d,d}^2\right\}} = \min\left\{D_{1,1}^{-2}, \ldots, D_{d,d}^{-2}\right\} = W_{\min}^2.$$

Hence,

$$\begin{aligned} W_{\min}^{-1/2} \left\|A^T(\mathbf{x} - \mu)\right\|^2 &= \left\|W_{\min}^{-1} A^T (\mathbf{x} - \mu)\right\|^2 \\ &= \left\|D^T U^T (\mathbf{x} - \mu)\right\|^2 = \left\|V D^T U^T (\mathbf{x} - \mu)\right\|^2 \qquad (1) \\ &= \left\|L^T(\mathbf{x} - \mu)\right\|^2 = (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu). \end{aligned}$$

We denote by I the identity matrix. We have that $I - A^TA = I - W_{\min}^2 D^2$ is a semi-positive definite matrix. Hence, there is a Cholesky decomposition $M^TM = I - A^TA$ of $I - A^TA$. Let $S^T = [A^T \mid M^T] \in \mathbb{R}^{d\times 2d}$, and let s denote a d-dimensional subspace of $\mathbb{R}^{2d}$ that is orthogonal to every column of S. Since $S^TS = A^TA + M^TM = I$, the columns of S are mutually orthogonal unit vectors. Hence,

$$\left\|A^T(\mathbf{x}-\mu)\right\| = \left\|S^T(\tilde{\mathbf{x}}-\tilde{\mu})\right\| = \left\|SS^T(\tilde{\mathbf{x}}-\tilde{\mu})\right\| = \mathrm{dist}(\tilde{\mathbf{x}}-\tilde{\mu}, s).$$

Together with (1) we obtain

$$W_{\min}^{-1/2}\mathrm{dist}(\tilde{\mathbf{x}}-\tilde{\mu}, s)^2 = W_{\min}^{-1/2}\left\|A^T(\mathbf{x}-\mu)\right\|^2 = (\mathbf{x}-\mu)^T\Sigma^{-1}(\mathbf{x}-\mu).$$

□

Remark A.2. *For the case that the smallest singular value $W_{\min}$ of L is at least 1, we have*

$$(\mathbf{x}-\mu)^T L^T L(\mathbf{x}-\mu) = \mathrm{dist}(\tilde{\mathbf{x}}-\tilde{\mu}, s)^2,$$

by replacing $W_{\min}$ with 1 in the above proof.

Theorem A.3 (Jensen's inequality)**.** *For every convex function $g : \mathbb{R} \to \mathbb{R}$ and a random variable X, we have $g(E[X]) \geq E[g(X)]$.*

Lemma A.4. *For every $y = (s, w, \tilde{\mu}, W) \in Y$, $0 < \varepsilon < 1/2$, and $\tilde{\mathbf{x}}, \tilde{\mathbf{x}}' \in \mathbb{R}^{2d}$, we have*

$$f_{\tilde{\mathbf{x}}}(y) \leq (1+2\varepsilon) f_{\tilde{\mathbf{x}}'}(y) + \frac{3W_{\max}\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')^2}{\varepsilon},$$

where $W_{\max} = \max_{1\leq i\leq k} W_i$.

Proof. Put $y = (s, w, \tilde{\mu}, W) \in Y$, $\tilde{\mathbf{x}}, \tilde{\mathbf{x}}' \in \mathbb{R}^{2d}$ and $1 \leq i \leq k$. By the triangle inequality,

$$\begin{aligned}|\mathrm{dist}(\tilde{\mathbf{x}}-\tilde{\mu}_i, s_i)^2 - \mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2| &= |\mathrm{dist}(\tilde{\mathbf{x}}-\tilde{\mu}_i, s_i) - \mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)|(\mathrm{dist}(\tilde{\mathbf{x}}-\tilde{\mu}_i, s_i) + \mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i))\\ &\leq \mathrm{dist}(\tilde{\mathbf{x}}-\tilde{\mu}_i, \tilde{\mathbf{x}}'-\tilde{\mu}_i)(2\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, x_i) + \mathrm{dist}(\tilde{\mathbf{x}}-\tilde{\mu}_i, \tilde{\mathbf{x}}'-\tilde{\mu}_i))\\ &= \mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')^2 + 2\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i).\end{aligned} \tag{2}$$

If $\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}') \leq \varepsilon\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)$ we have $\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i) \leq \varepsilon\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2$. Otherwise, $\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i) < \mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')/\varepsilon$ so $\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i) < \mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')^2/\varepsilon$. In both cases we thus obtain

$$\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i) \leq \mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')^2/\varepsilon + \varepsilon\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2.$$

Together with (2) we have

$$\begin{aligned}|\mathrm{dist}(\tilde{\mathbf{x}}-\tilde{\mu}_i, s_i)^2 - \mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2| &\leq \mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')^2 + 2\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')^2/\varepsilon + 2\varepsilon\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2\\ &= 3\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')^2/\varepsilon + 2\varepsilon\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2.\end{aligned} \tag{3}$$

Let $W_{\max} = \max_{1\leq i\leq k} W_i$. Using (3),

$$\begin{aligned}f_{\tilde{\mathbf{x}}}(y) &= -\ln\left(\sum_{i=1}^k w_i \exp\left(-W_i\mathrm{dist}(\tilde{\mathbf{x}}-\tilde{\mu}_i, s_i)^2\right)\right)\\ &\leq -\ln\left(\sum_{i=1}^k w_i \exp\left(-W_i\big(\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2 + 3\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')/\varepsilon + 2\varepsilon\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2\big)\right)\right)\\ &= -\ln\left(\sum_{i=1}^k w_i \exp\left(-W_i(1+2\varepsilon)\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2 - 3W_i\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')/\varepsilon\right)\right)\\ &\leq -\ln\left(\sum_{i=1}^k w_i \exp\left(-W_i(1+2\varepsilon)\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2\right) + 3W_{\max}\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')/\varepsilon\right)\\ &= -\ln\left(\sum_{i=1}^k w_i \left(\exp\left(-W_i\mathrm{dist}(\tilde{\mathbf{x}}'-\tilde{\mu}_i, s_i)^2\right)\right)^{1+2\varepsilon}\right) + 3W_{\max}\mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')/\varepsilon.\end{aligned} \tag{4}$$

Let X be a random variable such that $X = \exp\left(-W_i \mathrm{dist}(\tilde{\mathbf{x}}' - \tilde{\mu}_i, s_i)^2\right)$ with probability w_i, $1 \leq i \leq k$. Since $z^{1+2\varepsilon}$ is a convex function over $z \in [0,1]$, applying Theorem A.3 with $g(z) = z^{1+2\varepsilon}$ yields $\mathrm{E}[g(X)] \geq g(\mathrm{E}[X])$. Hence,

$$\sum_{i=1}^{k} w_i \left(\exp\left(-W_i \mathrm{dist}(\tilde{\mathbf{x}}' - \tilde{\mu}_i, s_i)^2\right)\right)^{1+2\varepsilon} = \mathrm{E}[X^{1+2\varepsilon}] = E[g(X)] \geq g(E[X]) = (E[X])^{1+2\varepsilon}$$
$$= \left(\sum_{i=1}^{k} w_i \exp\left(-W_i \mathrm{dist}(\tilde{\mathbf{x}}' - \tilde{\mu}_i, s_i)^2\right)\right)^{1+2\varepsilon}.$$

Combining the last inequality with (4) yields

$$\begin{aligned}
f_{\tilde{\mathbf{x}}}(y) &\leq -\ln\left(\sum_{i=1}^{k} w_i \left(\exp\left(-W_i \mathrm{dist}(\tilde{\mathbf{x}}' - \tilde{\mu}_i, s_i)^2\right)\right)^{1+2\varepsilon}\right) + 3W \mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')/\varepsilon \\
&\leq -\ln\left(\sum_{i=1}^{k} w_i \exp\left(-W_i \mathrm{dist}(\tilde{\mathbf{x}}' - \tilde{\mu}_i, s_i)^2\right)\right)^{1+2\varepsilon} + 3W_{\max} \mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')/\varepsilon \\
&= -(1+2\varepsilon)\ln\left(\sum_{i=1}^{k} w_i \exp\left(-W_i \mathrm{dist}(\tilde{\mathbf{x}}' - \tilde{\mu}_i, s_i)^2\right)\right) + 3W_{\max} \mathrm{dist}(p, p')/\varepsilon \\
&= (1+2\varepsilon) f_{\tilde{\mathbf{x}}'}(y) + 3W_{\max} \mathrm{dist}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}')/\varepsilon.
\end{aligned}$$

□

Lemma A.5. *Let y^* be a tuple that minimizes $\sum_{\tilde{\mathbf{x}} \in P} \mathrm{dist}(\tilde{\mathbf{x}}, y)^2$ over every $y \in Y$. Then for every $y \in Y$*

$$\sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, y^*)^2 \leq \sum_{\tilde{\mathbf{x}} \in \tilde{D}} f_{\tilde{\mathbf{x}}}(y).$$

Proof. Fix $y = (s, w, \tilde{\mu}, W) \in Y$ and $\tilde{\mathbf{x}} \in \tilde{D}$. We have

$$\begin{aligned}
f_{\tilde{\mathbf{x}}}(y) &= -\ln\left(\sum_{i=1}^{k} w_i \exp\left(-W_i \mathrm{dist}(\tilde{\mathbf{x}} - \tilde{\mu}_i, s_i)^2\right)\right) \\
&\geq -\ln\left(\sum_{i=1}^{k} w_i \cdot \exp\left(\mathrm{dist}(\tilde{\mathbf{x}}, y)^2\right)\right) = \mathrm{dist}(\tilde{\mathbf{x}}, y)^2.
\end{aligned}$$

Hence,

$$\sum_{\tilde{\mathbf{x}} \in \tilde{D}} f_{\tilde{\mathbf{x}}}(y) \geq \sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, s_m)^2 \geq \sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, y^*)^2$$

□

B Coresets For Semi-Spherical Mixtures of Gaussians

Let $y^* \in Y$ be a tuple that minimizes $\sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, y)$ over every $y \in Y$. Suppose that $B = \left\{b_1, b_2, \cdots, b_{|B|}\right\} \subseteq \mathbb{R}^{2d}$ and $\alpha > 0$ such that

$$\sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2 \leq \alpha \sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, y^*). \qquad (5)$$

Let $\left\{\tilde{D}_1, \cdots, \tilde{D}_{|B|}\right\}$ be a partition of $\tilde{D}$, where $\tilde{D}_i = \left\{\tilde{\mathbf{x}} \in \tilde{D} \mid \mathrm{dist}(\tilde{\mathbf{x}}, b_i) = \mathrm{dist}(\tilde{\mathbf{x}}, B)\right\}$ are the points that are served by b_i, $1 \leq i \leq |B|$. Ties are broken arbitrarily.

Lemma B.1. *For every $1 \le i \le |B|$, $\tilde{\mathbf{x}} \in \tilde{D}_i$ and $y = (s, w, \tilde{\mu}, W) \in Y$, we have*

$$\frac{f_{\tilde{\mathbf{x}}}(y)}{\sum_{\tilde{\mathbf{x}} \in \tilde{D}} f_{\tilde{\mathbf{x}}}(y)} \le \frac{O(1)(\varepsilon + \alpha W_{\max})}{\varepsilon |\tilde{\mathbf{x}}_i|} + \frac{O(1)\mathrm{dist}(\tilde{\mathbf{x}}, B)^2}{\varepsilon \sum_{\tilde{\mathbf{x}} \in \tilde{\mathbf{x}}} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2}.$$

Proof. Put $y = (s, w, \tilde{\mu}, W) \in Y$, $1 \le i \le |B|$, $\tilde{\mathbf{x}} \in \tilde{D}_i$, $n_i = |\tilde{D}_i|$, and $b = b_i$. By Lemma A.4,

$$f_{\tilde{\mathbf{x}}}(y) \le (1 + 2\varepsilon) f_b(y) + \frac{O(1) W_{\max} \mathrm{dist}(\tilde{\mathbf{x}}, b)^2}{\varepsilon} \le 2 f_b(y) + \frac{O(1) W_{\max} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2}{\varepsilon}. \tag{6}$$

We now bound $f_b(y)$. Applying Lemma A.4 with $\tilde{\mathbf{x}} \in \tilde{D}_i$ and b yields

$$f_b(y) \le (1 + 2\varepsilon) f_{\tilde{\mathbf{x}}}(y) + \frac{O(1) W_{\max} \mathrm{dist}(\tilde{\mathbf{x}}, b)^2}{\varepsilon}.$$

Summing over every $\tilde{\mathbf{x}} \in \tilde{D}_i$ yields

$$\begin{aligned} n_i \cdot f_b(y) &\le (1 + 2\varepsilon) \sum_{\tilde{\mathbf{x}} \in \tilde{D}_i} f_{\tilde{\mathbf{x}}}(y) + \frac{O(1) W_{\max} \sum_{\tilde{\mathbf{x}} \in \tilde{D}_i} \mathrm{dist}(\tilde{\mathbf{x}}, b)^2}{\varepsilon} \\ &\le 2 \sum_{\tilde{\mathbf{x}} \in \tilde{\mathbf{x}}} f_{\tilde{\mathbf{x}}}(y) + \frac{O(1) W_{\max} \sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2}{\varepsilon} \end{aligned}$$

Plugging the last inequality in (6) yields

$$f_{\tilde{\mathbf{x}}}(y) \le \frac{2}{n_i} \sum_{\tilde{\mathbf{x}} \in \tilde{D}} f_{\tilde{\mathbf{x}}}(y) + \frac{O(1) W_{\max} \sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2}{\varepsilon n_i} + \frac{O(1) W_{\max} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2}{\varepsilon}.$$

Hence,

$$\frac{f_{\tilde{\mathbf{x}}}(y)}{\sum_{\tilde{\mathbf{x}} \in \tilde{D}} f_{\tilde{\mathbf{x}}}(y)} \le \frac{2}{n_i} + \frac{O(1) W_{\max} \sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2}{\varepsilon n_i \sum_{\tilde{\mathbf{x}} \in \tilde{\mathbf{x}}} f_{\tilde{\mathbf{x}}}(y)} + \frac{O(1) W_{\max} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2}{\varepsilon \sum_{\tilde{\mathbf{x}} \in \tilde{D}} f_{\tilde{\mathbf{x}}}(y)}. \tag{7}$$

By (5) and Lemma A.5,

$$\sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2 \le \alpha \sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, y^*)^2 \le \alpha \sum_{\tilde{\mathbf{x}} \in \tilde{D}} f_{\tilde{\mathbf{x}}}(y).$$

Using the last inequality with (7) yields

$$\frac{f_{\tilde{\mathbf{x}}}(y)}{\sum_{\tilde{\mathbf{x}} \in \tilde{D}} f_{\tilde{\mathbf{x}}}(y)} \le \frac{2\varepsilon + O(1) \cdot W_{\max} \alpha}{\varepsilon n_i} + \frac{O(1) W_{\max} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2}{\varepsilon \sum_{\tilde{\mathbf{x}} \in \tilde{D}} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2}.$$

□

Theorem B.2 ([4])**.** *Let F be a set of n functions from Y to $[0, \infty)$ and $0 < \varepsilon < 1/4$. Let $m : F \to \mathcal{N} \setminus \{0\}$ be a function on F such that*

$$m(f) \ge n \cdot \max_{x \in X} \frac{f(x)}{\mathrm{cost}(F, x)}. \tag{8}$$

For each $f \in F$, let $g_f : Y \to [0, \infty)$ be defined as $g_f(y) = f(y)/m(f)$. Let G_f consists of m_f copies of g_f, and let S be a random sample of

$$t = \frac{\dim(F) \left(\sum_{f \in F} m(f)\right)^2}{(\varepsilon n)^2}$$

functions from the set $G = \bigcup_{f \in F} G_f$. Then for every $y \in Y$,

$$|\sum_{f \in F} f(y) - \sum_{f \in S} f(y)| \le \varepsilon \sum_{f \in F} f(y).$$

Theorem B.3 ([4])**.** *Let $B \subseteq \mathbb{R}^d$ be the set that is computed during the execution of the algorithm 1. Then*

$$\sum_{\mathbf{x} \in D} \mathrm{dist}(\mathbf{x}, B) \le O(1) \min_{C^* \subseteq \mathbb{R}^d, |C^*| = k} \sum_{\mathbf{x} \in D} \mathrm{dist}(\mathbf{x}, C^*)$$

C Complexity of Mixture of Gaussians

Definition C.1 (range space [24])**.** *A* range space *is a pair* $(F, \mathbf{ranges})$ *where* F *is a set, and* $\mathbf{ranges}$ *is a set of subsets of* F*. The* dimension *of the range space* $(F, \mathbf{ranges})$ *is the smallest integer* d*, such that for every* $G \subseteq F$ *we have*

$$\Big|\{G \cap \mathbf{range} \mid \mathbf{range} \in \mathbf{ranges}\}\Big| \le |G|^d \enspace .$$

The dimension of a range space relates (but is not equivalent) to a term known as the VC-dimension of a range space.

Definition C.2 (range space and dimension of F [25])**.** *Let* F *be a finite set of functions from a set* X *to* $[0,\infty)$*. The* dimension $\dim(F)$ *of* F *is the dimension of the range space* $\big(F, \mathbf{ranges}(F)\big)$*, where* $\mathbf{ranges}(F)$ *is the* range space of F*, that is defined as follows. For every* $x \in X$ *and* $r \ge 0$*, let* $\mathbf{range}(F,x,r) = \{f \in F \mid f(x) \le r\}$*. Let* $\mathbf{ranges}(F) = \{\mathbf{range}(F,x,r) \mid x \in X, r \ge 0\}$*.*

Theorem C.3 ([26])**.** *Let* q *be a natural number and suppose that* G *is the class of real-values functions in the variables* $y_1, \cdots, y_d$ *satisfying the following conditions: For every* $f \in G$ *there is affine functions* $g_1, \cdots, g_r$ *where* $r \le q$*, in the variables* $y_1, \cdot, y_d$ *such that* f *is an affine combination of* $y_1, \cdots, y_d$ *and* $e^{g_1}, \cdots, e^{g_r}$*. Then* G *has solution set components bound*

$$|B| = 2^{O(d^2q^2)}.$$

Theorem C.4.

$$\dim(Y) = O(kd^2).$$

Proof. Let $y = (s, \tilde{\mu}, w) \in Y$. Since s is a hyperplane, there is a tuple of $d+1$ vectors $h_0, \cdots, h_{d+1} \in \mathbb{R}^d$ such that $s = \Big\{h_0 + \sum_{i=1}^d a_i h_i \mid a_1, \ldots, a_d \in \mathbb{R}\Big\}$, and

$$\begin{aligned}\mathrm{dist}^2(\mathbf{x}, s) &= \|\mathbf{x} - h_0\|_2^2 - \sum_{i=1}^d \big((\mathbf{x} - h_0)^T h_i\big)^2 \\ &= \|\mathbf{x} - h_0\|_2^2 - \sum_{i=1}^d (\mathbf{x}^T h_i - h_0^T h_i)^2.\end{aligned}$$

For two vectors $(m_1, \cdots, m_s) \in \mathbb{R}^s$ and $(y_1, \cdots, y_t) \in \mathbb{R}^t$, we denote by my the tuple $m_1, \cdots, m_s, y_1, \cdots, y_t$. Let $h' = (1, h_0 \cdots h_d) \in \mathbb{R}^{d(d+1)+1}$, and $\mathbf{x}' = (1, \mathbf{x}) \in \mathbb{R}^{d+1}$. Hence, we have

$$\mathrm{dist}^2(\mathbf{x}, s) = \sum_{i_0, i_1 \in [d+1], i_2, i_3 \in [d(d+1)+1]} c_{i_0,i_1,i_2,i_3} p'_{i_0} \mathbf{x}'_{i_1} h'_{i_2} h'_{i_3}, \tag{9}$$

where c_{i_0,i_1,i_2,i_3} is a constant that depends only on $i_0, \ldots, i_3$, and equals to zero for all except $d_1 = O(d^2)$ terms of the summation. That is, $\mathrm{dist}^2(\mathbf{x}, \cdot)$ is an affine function . □

Theorem C.5. *Let* $D \subseteq \mathbb{R}^d$*,* $\delta, \varepsilon > 0$*,* $k \ge 1$ *and* $W_{\min}, W_{\max} > 0$*. Let* $\mathfrak{C}$ *be the collection of all mixtures of Gaussians* $\theta = [(w_1, \mu_1, \Sigma_1), \ldots, (w_k, \mu_k, \Sigma_k)]$ *such that* Σ_i *is a* $d \times d$ *covariance matrix whose singular values are between* $W_{\min}^2$ *and* $W_{\max}^2$*, for every* $1 \le i \le k$*. Let* C *be the output of Algorithm 1.*

Then, with probability at least $1 - \delta$*, for every mixture* $\theta \in \mathfrak{C}$ *we have*

$$\left|\sum_{\mathbf{x} \in D} \ln \sum_{i=1}^k w_i \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i)\right) - \sum_{\mathbf{x} \in C} \gamma(\mathbf{x}) \ln \sum_{i=1}^k w_i \exp\left(-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i)\right)\right|$$
$$\le \varepsilon \frac{W_{\max}^2}{W_{\min}^2} \mathcal{L}(D \mid \theta)$$

Proof. For every $\mathbf{x} \in D$ define $\tilde{\mathbf{x}} = \tilde{\mathbf{x}}(\mathbf{x}) = (\mathbf{x}, 0, \ldots, 0) \in \mathbb{R}^{2d}$ and let $\tilde{D} = \{\tilde{\mathbf{x}}(\mathbf{x}) \mid \mathbf{x} \in D\}$. Fix $\theta = [(w_1, \mu_1, \Sigma_1), \ldots, (w_k, \mu_k, \Sigma_k)] \in \mathfrak{C}$. For every i, $1 \le i \le k$, let W_i be the inverse of the smallest singular value of Σ_i. By Lemma A.1 there is a vector $\tilde{\mu}_i$, and a subspace s_i such that

$$-\frac{1}{2}(\mathbf{x} - \mu_i)^T \Sigma_i^{-1} (\mathbf{x} - \mu_i) = W_i \mathrm{dist}(\tilde{\mathbf{x}} - \tilde{\mu}_i, s_i)^2.$$

Let $y = y(\theta) = (s, w, \tilde{\mu}, W)$ where $\tilde{\mu} = (\tilde{\mu}_1, \cdots, \tilde{\mu}_k)$, $w = (w_1, \cdots, w_k)$, $W = (W_1, \cdots, W_k)$ and $s = (s_1, \cdots, s_k)$, $Y = \{y(\theta) \mid y \in Y\}$. For every $\tilde{\mathbf{x}} \in \tilde{D}$ and $y = (s, w, \tilde{\mu}, W) \in Y$ we define

$$f_{\tilde{\mathbf{x}}}(y) = -\ln\left(\sum_{i=1}^{k} w_i \exp\left(-W_i \mathrm{dist}(\tilde{\mathbf{x}} - \tilde{\mu}_i, s_i)^2\right)\right),$$

and $F = \left\{f_{\tilde{\mathbf{x}}} \mid \tilde{\mathbf{x}} \in \tilde{D}\right\}$. By Theorem C.4, we have $\dim(F) = O(d^2 k)$. For every $f \in F$, let

$$m(f) = \frac{O(1)(\varepsilon + \alpha W_{\max})}{\varepsilon |\tilde{\mathbf{x}}_i|} + \frac{O(1)\mathrm{dist}(\tilde{\mathbf{x}}, B)^2}{\varepsilon \sum_{\tilde{\mathbf{x}} \in \tilde{\mathbf{x}}} \mathrm{dist}(\tilde{\mathbf{x}}, B)^2},$$

and note that $\sum_{f \in F} m(f) = O(|B|)$. By Lemma B.1, for every $f \in F$ we have

$$m(f) \geq n \cdot \max_{y \in Y} \frac{f(y)}{\sum_{f \in F} f(y)}.$$

Let G_f consists of m_f copies of g_f, and note that C in Algorithm 1 is a uniform random sample from G_F. By Theorem B.2 we thus have

$$\left|\sum_{f \in F} f(y) - \sum_{f \in C} f(y)\right| \leq \varepsilon \sum_{f \in F} f(y).$$

□

Observation C.6. *Let $\theta = [(w_1, \mu_1, \Sigma_1), \ldots, (w_k, \mu_k, \Sigma_k)] \in \mathfrak{C}$ such that*

$$\forall_i : \left(\prod_{\lambda \in \mathrm{spec}(\Sigma_i)} \lambda\right) \geq \frac{1}{(2\pi)^d}.$$

If C is a (k, ε)-coreset for $D \subseteq \mathbb{R}^d$, then $\mathcal{L}(C \mid \theta)$ is a $(1+\varepsilon)$-approximation for $\mathcal{L}(D \mid \theta)$. That is,

$$(1 - \varepsilon)\mathcal{L}(D \mid \theta) \leq \mathcal{L}(C \mid \theta) \leq \mathcal{L}(D \mid \theta)(1 + \varepsilon).$$

Proof. Since $\sum_i w_i = 1$, we have by the assumption on θ

$$Z(\theta) = \sum_i \frac{w_i}{\sqrt{|2\pi\Sigma_i|}} \leq \max_i \frac{1}{\sqrt{|2\pi\Sigma_i|}} = \max_i \frac{1}{\sqrt{(2\pi)^d \prod_{\lambda \in \mathrm{spec}(\Sigma_i)} \lambda}} \leq 1.$$

Hence, $-n \ln Z(\theta) \geq 0$. By this and the assumption on θ,

$$\begin{aligned}
\mathcal{L}(C \mid \theta) &= -n \ln Z(\theta) + \phi(C \mid \theta) \\
&\leq -n \ln Z(\theta) + (1 + \varepsilon)\phi(D \mid \theta) \\
&\leq (1 + \varepsilon)\big(-n \ln Z(\theta) + \phi(D \mid \theta)\big) \\
&= (1 + \varepsilon)\mathcal{L}(D \mid \theta).
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathcal{L}(C \mid \theta) &= -n \ln Z(\theta) + \phi(C \mid \theta) \\
&\geq -n \ln Z(\theta) + (1 - \varepsilon)\phi(D \mid \theta) \\
&\geq (1 - \varepsilon)\big(-n \ln Z(\theta) + \phi(D \mid \theta)\big) \\
&= (1 - \varepsilon)\mathcal{L}(D \mid \theta).
\end{aligned}$$

□