Using multiple samples to learn mixture models

Jason Lee*
Stanford University
Stanford, USA
jdl17@stanford.edu

Ran Gilad-Bachrach
Microsoft Research
Redmond, USA
rang@microsoft.com

Rich Caruana
Microsoft Research
Redmond, USA
rcaruana@microsoft.com

Abstract

In the mixture models problem it is assumed that there are K distributions $\theta_1, \ldots, \theta_K$ and one gets to observe a sample from a mixture of these distributions with unknown coefficients. The goal is to associate instances with their generating distributions, or to identify the parameters of the hidden distributions. In this work we make the assumption that we have access to several samples drawn from the same K underlying distributions, but with different mixing weights. As with topic modeling, having multiple samples is often a reasonable assumption. Instead of pooling the data into one sample, we prove that it is possible to use the differences between the samples to better recover the underlying structure. We present algorithms that recover the underlying structure under milder assumptions than the current state of art when either the dimensionality or the separation is high. The methods, when applied to topic modeling, allow generalization to words not present in the training data.

1 Introduction

The mixture model has been studied extensively from several directions. In one setting it is assumed that there is a single sample, that is a single collection of instances, from which one has to recover the hidden information. A line of studies on clustering theory, starting from [5] has proposed to address this problem by finding a projection to a low dimensional space and solving the problem in this space. The goal of this projection is to reduce the dimension while preserving the distances, as much as possible, between the means of the underlying distributions. We will refer to this line as MM (Mixture Models). On the other end of the spectrum, Topic modeling (TM), [9, 3], assumes multiple samples (documents) that are mixtures, with different weights of the underlying distributions (topics) over words.

Comparing the two lines presented above shows some similarities and some differences. Both models assume the same generative structure: a point (word) is generated by first choosing the distribution θ_i using the mixing weights and then selecting a point (word) according to this distribution. The goal of both models is to recover information about the generative model (see [10] for more on that). However, there are some key differences:

(a) In MM, there exists a single sample to learn from. In TM, each document is a mixture of the topics, but with different mixture weights.

(b) In MM, the points are represented as feature vectors while in TM the data is represented as a word-document co-occurrence matrix. As a consequence, the model generated by TM cannot assign words that did not previously appear in any document to topics.

*Work done while the author was an intern at Microsoft Resaerch

(c) TM assumes high density of the samples, i.e., that the each word appears multiple times. However, if the topics were not discrete distributions, as is mostly the case in MM, each "word" (i.e., value) would typically appear either zero or one time, which makes the co-occurrence matrix useless.

In this work we try to close the gap between MM and TM. Similar to TM, we assume that multiple samples are available. However, we assume that points (words) are presented as feature vectors and the hidden distributions may be continuous. This allows us to solve problems that are typically hard in the MM model with greater ease and generate models that generalize to points not in the training data which is something that TM cannot do.

1.1 Definitions and Notations

We assume a mixture model in which there are K mixture components $\theta_1, \ldots, \theta_K$ defined over the space X. These mixture components are probability measures over X. We assume that there are M mixture models (samples), each drawn with different mixture weights $\Phi^1, \ldots, \Phi^M$ such that $\Phi^j = (\phi_1^j, \ldots, \phi_K^j)$ where all the weights are non-negative and sum to 1. Therefore, we have M different probability measures $D_1, \ldots, D_M$ defined over X such that for a measurable set A and $j = 1, \ldots, M$ we have $D_j(A) = \sum_i \phi_i^j \theta_i(A)$. We will denote by $\phi_{\min}$ the minimal value of ϕ_i^j.

In the first part of this work, we will provide an algorithm that given samples $S_1, \ldots, S_M$ from the mixtures $D_1, \ldots, D_M$ finds a low-dimensional embedding that preserves the distances between the means of each mixture.

In the second part of this work we will assume that the mixture components have disjoint supports. Hence we will assume that $X = \cup_j C_j$ such that the C_j's are disjoint and for every j, $\theta_j(C_j) = 1$. Given samples $S_1, \ldots, S_M$, we will provide an algorithm that finds the supports of the underlying distributions, and thus clusters the samples.

1.2 Examples

Before we dive further in the discussion of our methods and how they compare to prior art, we would like to point out that the model we assume is realistic in many cases. Consider the following example: assume that one would like to cluster medical records to identify sub-types of diseases (e.g., different types of heart disease). In the classical clustering setting (MM), one would take a sample of patients and try to divide them based on some similarity criteria into groups. However, in many cases, one has access to data from different hospitals in different geographical locations. The communities being served by the different hospitals may be different in socioeconomic status, demographics, genetic backgrounds, and exposure to climate and environmental hazards. Therefore, different disease sub-types are likely to appear in different ratios in the different hospital. However, if patients in two hospitals acquired the same sub-type of a disease, parts of their medical records will be similar.

Another example is object classification in images. Given an image, one may break it to patches, say of size 10x10 pixels. These patches may have different distributions based on the object in that part of the image. Therefore, patches from images taken at different locations will have different representation of the underlying distributions. Moreover, patches from the center of the frame are more likely to contain parts of faces than patches from the perimeter of the picture. At the same time, patches from the bottom of the picture are more likely to be of grass than patches from the top of the picture.

In the first part of this work we discuss the problem of identifying the mixture component from multiple samples when the means of the different components differ and variances are bounded. We focus on the problem of finding a low dimensional embedding of the data that preserves the distances between the means since the problem of finding the mixtures in a low dimensional space has already been address (see, for example [10]). Next, we address a different case in which we assume that the support of the hidden distributions is disjoint. We show that in this case we can identify the supports of each distribution. Finally we demonstrate our approaches on toy problems. The proofs of the theorems and lemmas

appear in the appendix. Table 1 summarizes the applicability of the algorithms presented here to the different scenarios.

1.3 Comparison to prior art

There are two common approaches in the theoretical study of the MM model. The method of moments [6, 8, 1] allows the recovery of the model but requires exponential running time and sample sizes. The other approach, to which we compare our results, uses a two stage approach. In the first stage, the data is projected to a low dimensional space and in the second stage the association of points to clusters is recovered. Most of the results with this approach assume that the mixture components are Gaussians. Dasgupta [5], in a seminal paper, presented the first result in this line. He used random projections to project the points to a space of a lower dimension. This work assumes that separation is at least $\Omega(\sigma_{\max}\sqrt{n})$. This result has been improved in a series of papers. Arora and Kannan [10] presented algorithms for finding the mixture components which are, in most cases, polynomial in n and K. Vempala and Wang [11] used PCA to reduce the required separation to $\Omega\left(\sigma_{\max}K^{1/4}\log^{1/4}\left(n/\phi_{\min}\right)\right)$. They use PCA to project on the first K principal components, however, they require the Gaussians to be spherical. Kanan, Salmasian and Vempala [7] used similar spectral methods but were able to improve the results to require separation of only $c\sigma_{\max}{}^{K^{2/3}}/\phi^2_{\min}$. Chaudhuri [4] have suggested using correlations and independence between features under the assumption that the means of the Gaussians differ on many features. They require separation of $\Omega\left(\sigma_{\max}\sqrt{K\log(K\sigma_{\max}\log n/\phi_{\min})}\right)$, however they assume that the Gaussians are axis aligned and that the distance between the centers of the Gaussians is spread across $\Omega\left(K\sigma_{\max}\log n/\phi_{\min}\right)$ coordinates.

	Disjoint clusters	**Overlapping clusters**
High dimension	DSC, MSP	MSP
Low dimension	DSC	

Table 1: Summary of the scenarios the MSP (Multi Sample Projection) algorithm and the DSC (Double Sample Clustering) algorithm are designed to address.

We present a method to project the problem into a space of dimension d^* which is the dimension of the affine space spanned by the means of the distributions. We can find this projection and maintain the distances between the means to within a factor of $1-\epsilon$. The different factors, $\sigma_{\max}$, n and ϵ will affect the sample size needed, but do not make the problem impossible. This can be used as a preprocessing step for any of the results discussed above. For example, combining with [5] yields an algorithm that requires a separation of only $\Omega\left(\sigma_{\max}\sqrt{d^*}\right) \leq \Omega\left(\sigma_{\max}\sqrt{K}\right)$. However, using [11] will result in separation requirement of $\Omega\left(\sigma_{\max}\sqrt{K\log\left(K\sigma_{\max}\log d^*/\phi_{\min}\right)}\right)$. There is also an improvement in terms of the value of $\sigma_{\max}$ since we need only to control the variance in the affine space spanned by the means of the Gaussians and do not need to restrict the variance in orthogonal directions, as long as it is finite. Later we also show that we can work in a more generic setting where the distributions are not restricted to be Gaussians as long as the supports of the distributions are disjoint. While the disjoint assumption may seem too strict, we note that the results presented above make very similar assumptions. For example, even if the required separation is $\sigma_{\max}K^{1/2}$ then if we look at the Voronoi tessellation around the centers of the Gaussians, each cell will contain at least $1-(2\pi)^{-1}K^{3/4}\exp\left(-K/2\right)$ of the mass of the Gaussian. Therefore, when K is large, the supports of the Gaussians are almost disjoint.

2 Projection for overlapping components

In this section we present a method to use multiple samples to project high dimensional mixtures to a low dimensional space while keeping the means of the mixture components

Algorithm 1 Multi Sample Projection (MSP)

Inputs:
Samples $S_1, \ldots, S_m$ from mixtures $D_1, \ldots, D_m$
Outputs:
Vectors $\bar{v}_1, \ldots, \bar{v}_{m-1}$ which span the projected space
Algorithm:

1. For $j = 1, \ldots, m$ let $\bar{E}_j$ be the mean of the sample S_j
2. For $j = 1, \ldots, m-1$ let $\bar{v}_j = \bar{E}_j - \bar{E}_{j+1}$
3. return $\bar{v}_1, \ldots, \bar{v}_{m-1}$

well separated. The main idea behind the Multi Sample Projection (MSP) algorithm is simple. Let μ_i be the mean of the i'th component θ_i and let E_j be the mean of the j'th mixture D_j. From the nature of the mixture, E_j is in the convex-hull of $\mu_1, \ldots, \mu_K$ and hence in the affine space spanned by them; this is demonstrated in Figure 1. Under mild assumptions, if we have sufficiently many mixtures, their means will span the affine space spanned by $\mu_1, \ldots, \mu_K$. Therefore, the MSP algorithm estimates the E_j's and projects to the affine space they span. The reason for selecting this sub-space is that by projecting on this space we maintain the distance between the means while reducing the dimension to at most $K-1$. The MSP algorithm is presented in Algorithm 1. In the following theorem we prove the main properties of the MSP algorithm. We will assume that $X = \mathbb{R}^n$, the first two moments of θ_j are finite, and $\sigma^2_{\max}$ denotes maximal variance of any of the components in any direction. The separation of the mixture components is $\min_{j \neq j'} \|\mu_j - \mu_{j'}\|$. Finally, we will denote by d^* the dimension of the affine space spanned by the μ_j's. Hence, $d^* \leq K-1$.

Theorem 1. ***MSP Analysis***

Let $E_j = E[D_j]$ and let $v_j = E_j - E_{j+1}$. Let $N_j = |S_j|$. The following holds for MSP:

1. *The computational complexity of the MSP algorithm is $n \sum_{j=1}^{M} N_j + 2n(m-1)$ where n is the original dimension of the problem.*
2. *For any $\epsilon > 0$, $\Pr\left[\sup_j \left\|E_j - \bar{E}_j\right\| > \epsilon\right] \leq \frac{n\sigma^2_{max}}{\epsilon^2} \sum_j \frac{1}{N_j}$.*
3. *Let $\bar{\mu}_i$ be the projection of μ_i on the space spanned by $\bar{v}_1, \ldots, \bar{v}_{M-1}$ and assume that $\forall i,\ \mu_i \in span\{v_j\}$. Let α^i_j be such that $\mu_i = \sum_j \alpha^i_j v_j$ and let $A = \max_i \sum |\alpha^i_j|$ then with probability of at least $1 - \frac{n\sigma^2_{max}}{\epsilon^2} \sum_j \frac{1}{N_j}$*

$$\Pr\left[\max_{i,i'} |\|\mu_i - \mu_{i'}\| - \|\bar{\mu}_i - \bar{\mu}_{i'}\|| > \epsilon\right] \leq \frac{4n\sigma^2_{max}A^2}{\epsilon^2} \sum_j \frac{1}{N_j} \ .$$

The MSP analysis theorem shows that with large enough samples, the projection will maintain the separation between the centers of the distributions. Moreover, since this is a projection, the variance in any direction cannot increase. The value of A measures the complexity of the setting. If the mixing coefficients are very different in the different samples then A will be small. However, if the mixing coefficients are very similar, a larger sample is required. Nevertheless, the size of the sample needed is polynomial in the parameters of the problem. It is also apparent that with large enough samples, a good projection will be found, even with

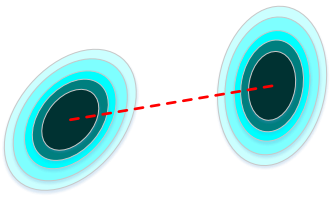


Figure 1: The mean of the mixture components will be in the convex hull of their means demonstrated here by the red line.

large variances, high dimensions and close centroids.

A nice property of the bounds presented here is that they assume only bounded first and second moments. Once a projection to a low dimensional space has been found, it is possible to find the clusters using approaches presented in section 1.3. However, the analysis of the MSP algorithm assumes that the means of $E_1, \ldots, E_M$ span the affine space spanned by $\mu_1, \ldots, \mu_K$. Clearly, this implies that we require that $m > d^*$. However, when m is much larger than d^*, we might end-up with a projection on too large a space. This could easily be fixed since in this case, $\bar{E}_1, \ldots, \bar{E}_m$ will be almost co-planar in the sense that there will be an affine space of dimension d^* that is very close to all these points and we can project onto this space.

3 Disjoint supports and the Double Sample Clustering (DSC) algorithm

In this section we discuss the case where the underlying distributions have disjoint supports. In this case, we do not make any assumption about the distributions. For example, we do not require finite moments. However, as in the mixture of Gaussians case some sort of separation between the distributions is needed, this is the role of the disjoint supports.

We will show that given two samples from mixtures with different mixture coefficients, it is possible to find the supports of the underlying distributions (clusters) by building a tree of classifiers such that each leaf represents a cluster. The tree is constructed in a greedy fashion. First we take the two samples, from the two distributions, and reweigh the examples such that the two samples will have the same cumulative weight. Next, we train a classifier to separate between the two samples. This classifier becomes the root of the tree. It also splits each of the samples into two sets. We take all the examples that the classifier assign to the label $+1(-1)$, reweigh them and train another classifier to separate between the two samples. We keep going in the same fashion until we can no longer find a classifier that splits the data significantly better than random.

To understand why this algorithm works it is easier to look first at the case where the mixture distributions are known. If D_1 and D_2 are known, we can define the L_1 distance between them as $L_1(D_1, D_2) = \sup_A |D_1(A) - D_2(A)|$.[1] It turns out that the supremum is attained by a set A such that for any i, $\mu_i(A)$ is either zero or one. Therefore, any inner node in the tree splits the region without breaking clusters. This process proceeds until all the points associated with a leaf are from the same cluster in which case, no classifier can distinguish between the classes.

When working with samples, we have to tolerate some error and prevent overfitting. One way to see that is to look at the problem of approximating the L_1 distance between D_1 and D_2 using samples S_1 and S_2. One possible way to do that is to define $\hat{L}_1 = \sup_A \left| \frac{A \cap S_1}{|S_1|} - \frac{A \cap S_2}{|S_2|} \right|$. However, this estimate is almost surely going to be 1 if the underlying distributions are absolutely continuous. Therefore, one has to restrict the class from which A can be selected to a class of VC dimension small enough compared to the sizes of the samples. We claim that asymptotically, as the sizes of the samples increase, one can increase the complexity of the class until the clusters can be separated.

Before we proceed, we recall a result of [2] that shows the relation between classification and the L_1 distance. We will abuse the notation and treat A both as a subset and as a classifier. If we mix D_1 and D_2 with equal weights then

$$\begin{aligned} \mathrm{err}(A) &= D_1(X \setminus A) + D_2(A) \\ &= 1 - D_1(A) + D_2(A) \\ &= 1 - (D_1(A) - D_2(A)) \quad . \end{aligned}$$

Therefore, minimizing the error is equivalent to maximizing the L_1 distance.

[1] the supremum is over all the measurable sets.

Algorithm 2 Double Sample Clustering (DSC)

Inputs:

- Samples S_1, S_2
- A binary learning algorithm L that given samples S_1, S_2 with weights w_1, w_2 finds a classifier h and an estimator e of the error of h.
- A threshold $\tau > 0$.

Outputs:

- A tree of classifiers

Algorithm:

1. Let $w_1 = 1$ and $w_2 = {}^{|S_1|}/_{|S_2|}$
2. Apply L to S_1 & S_2 with weights w_1 & w_2 to get the classifier h and estimator e.
3. If $e \geq \frac{1}{2} - \tau$,
   (a) return a tree with a single leaf.
4. else
   (a) For $j = 1, 2$, let $S_j^+ = \{x \in S_j \text{ s.t. } h(x) > 0\}$
   (b) For $j = 1, 2$, let $S_j^- = \{x \in S_j \text{ s.t. } h(x) < 0\}$
   (c) Let T^+ be the tree returned by the DSC algorithm applied to S_1^+ and S_2^+
   (d) Let T^- be the tree returned by the DSC algorithm applied to S_1^- and S_2^-
   (e) return a tree in which c is at the root node and T^- is its left subtree and T^+ is its right subtree

The key observation for the DSC algorithm is that if $\phi_i^1 \neq \phi_i^2$, then a set A that maximizes the L_1 distance between D_1 and D_2 is aligned with cluster boundaries (up to a measure zero). Furthermore, A contains all the clusters for which $\phi_i^1 > \phi_i^2$ and does not contain all the clusters for which $\phi_i^1 < \phi_i^2$. Hence, if we split the space to A and $\bar{A}$ we have few clusters in each side. By applying the same trick recursively in each side we keep on bisecting the space according to cluster boundaries until subspaces that contain only a single cluster remain. These sub-spaces cannot be further separated and hence the algorithm will stop. Figure 2 demonstrates this idea. The following lemma states this argument mathematically:

Figure 2: Demonstration of the DSC algorithm. Assume that $\Phi^1 = (0.4, 0.3, 0.3)$ for the orange, green and blue regions respectively and $\Phi^2 = (0.5, 0.1, 0.4)$. The green region maximizes the L_1 distance and therefore will be separated from the blue and orange. Conditioned on these two regions, the mixture coefficients are $\Phi^1_{\text{orange, blue}} = ({}^4/_7, {}^3/_7)$ and $\Phi^2_{\text{orange, blue}} = ({}^5/_9, {}^4/_9)$. The region that maximized this conditional L_1 is the orange regions that will be separated from the blue.

Lemma 1. *If* $D_j = \sum_i \phi_i^j \theta_i$ *then*

1. $L_1(D_1, D_2) \leq \sum_i \max\left(\phi_i^1 - \phi_i^2, 0\right)$.
2. *If* $A^* = \cup_{i:\phi_i^1 > \phi_i^2} C_i$ *then* $D_1(A^*) - D_2(A^*) = \sum_i \max\left(\phi_i^1 - \phi_i^2, 0\right)$.
3. *If* $\forall i,\ \phi_i^1 \neq \phi_i^2$ *and* A *is such that* $D_1(A) - D_2(A) = L_1(D_1, D_2)$ *then* $\forall i,\ \theta_i(A \Delta A^*) = 0$.

We conclude from Lemma 1 that if D_1 and D_2 were explicitly known and one could have found a classifier that best separates between the distributions, that classifier would not break clusters as long as the mixing coefficients

are not identical. In order for this to hold when the separation is applied recursively in the DSC algorithm it suffices to have that for every $I \subseteq [1, \ldots, K]$ if $|I| > 1$ and $i \in I$ then

$$\frac{\phi_i^1}{\sum_{i' \in I} \phi_{i'}^1} \neq \frac{\phi_i^2}{\sum_{i' \in I} \phi_{i'}^2}$$

to guarantee that at any stage of the algorithm clusters will not be split by the classifier (but may be sections of measure zero). This is also sufficient to guarantee that the leaves will contain single clusters.

In the case where data is provided through a finite sample then some book-keeping is needed. However, the analysis follows the same path. We show that with samples large enough, clusters are only minimally broken. For this to hold we require that the learning algorithm L separates the clusters according to this definition:

Definition 1. *For $I \subseteq [1, \ldots, K]$ let $c_I : X \mapsto \{\pm 1\}$ be such that $c_I(x) = 1$ if $x \in \cup_{i \in I} C_i$ and $c_I(x) = -1$ otherwise. A learning algorithm* L separates $C_1, \ldots, C_K$ *if for every $\epsilon, \delta > 0$ there exists N such that for every $n > N$ and every measure ν over $X \times \{\pm 1\}$ with probability $1 - \delta$ over samples from ν^n:*

1. *The algorithm L returns an hypothesis $h : X \mapsto \{\pm 1\}$ and an error estimator $e \in [0, 1]$ such that $\left| \Pr_{x,y \sim \nu} [h(x) \neq y] - e \right| \leq \epsilon$*
2. *h is such that*
$$\forall I, \quad \Pr_{x,y \sim \nu} [h(x) \neq y] < \Pr_{x,y \sim \nu} [c_I(x) \neq y] + \epsilon \ .$$

Before we introduce the main statement, we define what it means for a tree to cluster the mixture components:

Definition 2. *A* clustering tree *is a tree in which in each internal node is a classifier and the points that end in a certain leaf are considered a cluster. A clustering tree* ϵ-clusters *the mixture coefficient $\theta_1, \ldots, \theta_K$ if for every $i \in 1, \ldots, K$ there exists a leaf in the tree such that the cluster $L \subseteq X$ associated with this leaf is such that $\theta_i(L) \geq 1 - \epsilon$ and $\theta_{i'}(L) < \epsilon$ for every $i' \neq i$.*

To be able to find a clustering tree, the two mixtures have to be different. The following definition captures the gap which is the amount of difference between the mixtures.

Definition 3. *Let Φ^1 and Φ^2 be two mixture vectors. The* gap*, g, between them is*

$$g = \min \left\{ \left| \frac{\phi_i^1}{\sum_{i' \in I} \phi_{i'}^1} - \frac{\phi_i^2}{\sum_{i' \in I} \phi_{i'}^2} \right| : I \subseteq [1, \ldots, K] \text{ and } |I| > 1 \right\} \ .$$

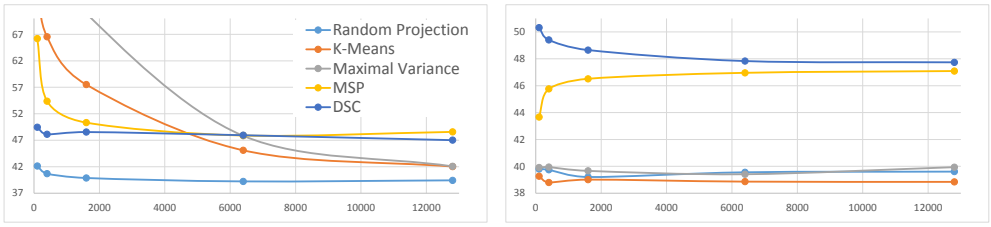
We say that Φ is b bounded away *from zero if $b \leq \min_i \phi_i$.*

Theorem 2. *Assume that* L *separates $\theta_1, \ldots, \theta_K$, there is a gap $g > 0$ between Φ^1 and Φ^2 and both Φ^1 and Φ^2 are bounded away from zero by $b > 0$. For every $\epsilon^*, \delta^* > 0$ there exists $N = N(\epsilon^*, \delta^*, g, b, K)$ such that given two random samples of sizes $N < n_1, n_2$ from the two mixtures, with probability of at least $1 - \delta^*$ the DSC algorithm will return a clustering tree which ϵ^*-clusters $\theta_1, \ldots, \theta_K$ when applied with the threshold $\tau = g/8$.*

4 Empirical evidence

We conducted several experiments with synthetic data to compare different methods when clustering in high dimensional spaces. The synthetic data was generated from three Gaussians with centers at points $(0, 0)$, $(3, 0)$ and $(-3, +3)$. On top of that, we added additional dimensions with normally distributed noise. In the first experiment we used unit variance for all dimensions. In the second experiment we skewed the distribution so that the variance in the other features is 5.

Two sets of mixing coefficients for the three Gaussians were chosen at random 100 times by selecting three uniform values from [0, 1] and normalizing them to sum to 1. We generated



(a) Accuracy with spherical Gaussians (b) Average accuracy with skewed Gaussians

Figure 3: **Comparison the different algorithms:** The dimension of the problem is presented in the X axis and the accuracy on the Y axis.

two samples with 80 examples each from the two mixing coefficients. The DSC and MSP algorithm received these two samples as inputs while the reference algorithms, which are not designed to use multiple samples, received the combined set of 160 points as input.

We ran 100 trials. In each trial, each of the algorithms finds 3 Gaussians. We then measure the percentage of the points associated with the true originating Gaussian after making the best assignment of the inferred centers to the true Gaussians.

We compared several algorithms. K-means was used on the data as a baseline. We compared three low dimensional projection algorithms. Following [5] we used random projections as the first of these. Second, following [11] we used PCA to project on the maximal variance subspace. MSP was used as the third projection algorithm. In all projection algorithm we first projected on a one dimensional space and then applied K-means to find the clusters. Finally, we used the DSC algorithm. The DSC algorithm uses the classregtree function in MATLAB as its learning oracle. Whenever K-means was applied, the MATLAB implementation of this procedure was used with 10 random initial starts.

Figure 3(a) shows the results of the first experiment with unit variance in the noise dimensions. In this setting, the Maximal Variance method is expected to work well since the first two dimensions have larger expected variance. Indeed we see that this is the case. However, when the number of dimensions is large, MSP and DSC outperform the other methods; this corresponds to the difficult regime of low signal to noise ratio. In 12800 dimensions, MSP outperforms Random Projections 90% of the time, Maximal Variance 80% of the time, and K-means 79% of the time. DSC outperforms Random Projections, Maximal Variance and K-means 84%, 69%, and 66% of the time respectively. Thus the p-value in all these experiments is < 0.01.

Figure 3(b) shows the results of the experiment in which the variance in the noise dimensions is higher which creates a more challanging problem. In this case, we see that all the reference methods suffer significantly, but the MSP and the DSC methods obtain similar results as in the previous setting. Both the MSP and the DSC algorithms win over Random Projections, Maximal Variance and K-means more than 78% of the time when the dimension is 400 and up. The p-value of these experiments is $< 1.6 \times 10^{-7}$.

5 Conclusions

The mixture problem examined here is closely related to the problem of clustering. Most clustering data can be viewed as points generated from multiple underlying distributions or generating functions, and clustering can be seen as the process of recovering the structure of or assignments to these distributions. We presented two algorithms for the mixture problem that can be viewed as clustering algorithms. The MSP algorithm uses multiple samples to find a low dimensional space to project the data to. The DSC algorithm builds a clustering tree assuming that the clusters are disjoint. We proved that these algorithms work under milder assumptions than currently known methods. The key message in this work is that when multiple samples are available, often it is best not to pool the data into one large sample, but that the structure in the different samples can be leveraged to improve clustering power.

References

[1] Mikhail Belkin and Kaushik Sinha, *Polynomial learning of distribution families*, Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on, IEEE, 2010, pp. 103–112.

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira, *Analysis of representations for domain adaptation*, Advances in neural information processing systems **19** (2007), 137.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan, *Latent dirichlet allocation*, the Journal of machine Learning research **3** (2003), 993–1022.

[4] Kamalika Chaudhuri and Satish Rao, *Learning mixtures of product distributions using correlations and independence*, Proc. of COLT, 2008.

[5] Sanjoy Dasgupta, *Learning mixtures of gaussians*, Foundations of Computer Science, 1999. 40th Annual Symposium on, IEEE, 1999, pp. 634–644.

[6] Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant, *Efficiently learning mixtures of two gaussians*, Proceedings of the 42nd ACM symposium on Theory of computing, ACM, 2010, pp. 553–562.

[7] Ravindran Kannan, Hadi Salmasian, and Santosh Vempala, *The spectral method for general mixture models*, Learning Theory, Springer, 2005, pp. 444–457.

[8] Ankur Moitra and Gregory Valiant, *Settling the polynomial learnability of mixtures of gaussians*, Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on, IEEE, 2010, pp. 93–102.

[9] Christos H Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala, *Latent semantic indexing: A probabilistic analysis*, Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems, ACM, 1998, pp. 159–168.

[10] Arora Sanjeev and Ravi Kannan, *Learning mixtures of arbitrary gaussians*, Proceedings of the thirty-third annual ACM symposium on Theory of computing, ACM, 2001, pp. 247–257.

[11] Santosh Vempala and Grant Wang, *A spectral algorithm for learning mixtures of distributions*, Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on, IEEE, 2002, pp. 113–122.

A Supplementary Material

Here we provide detailed analysis of the results presented in the paper that could not fit due to space limitations.

A.1 Proof of Lemma 1

Proof. 1. Let A be a measurable set let $A_i = A \cap C_i$.

$$\begin{aligned} D_1(A) - D_2(A) &= \sum_i D_1(A_i) - D_2(A_i) \\ &= \sum_i \phi_i^1 \theta_i(A_i) - \phi_i^2 \theta_i(A_i) \\ &= \sum_i \theta_i(A_i)\left(\phi_i^1 - \phi_i^2\right) \\ &\leq \sum_i \max\left(\phi_i^1 - \phi_i^2, 0\right) \quad . \end{aligned}$$

2. Let $A^* = \cup_{i:\phi_i^1 > \phi_i^2} C_i$ then

$$\begin{aligned} D_1(A^*) - D_2(A^*) &= \sum_{i:\phi_i^1 > \phi_i^2} \theta_i(C_i)\left(\phi_i^1 - \phi_i^2\right) \\ &= \sum_i \max\left(\phi_i^1 - \phi_i^2, 0\right) \quad . \end{aligned}$$

3. Assume that $\forall i,\ \phi_i^1 \neq \phi_i^2$ and A is such that $D_1(A) - D_2(A) = L_1(D_1, D_2)$. As before let $A_i = A \cap C_i$ then

$$D_1(A) - D_2(A) = \sum_i \theta_i(A_i)\left(\phi_i^1 - \phi_i^2\right) \quad .$$

If exists i such that $\theta_i(A \Delta A^*) \neq 0$ then there could be two cases. If i^* is such that $\phi_{i^*}^1 > \phi_{i^*}^2$ then $\theta_{i^*}(A^*) = 1$ hence $\theta_{i^*}(A) < 1$. Therefore,

$$\begin{aligned} D_1(A) - D_2(A) &\leq \sum_i \theta_i(A_i) \max\left(\phi_i^1 - \phi_i^2, 0\right) \\ &< \sum_i \max\left(\phi_i^1 - \phi_i^2, 0\right) \end{aligned}$$

which contradicts the assumptions. In the same way, if i^* is such that $\phi_{i^*}^1 < \phi_{i^*}^2$ then $\theta_{i^*}(A^*) = 0$ hence $\theta_{i^*}(A) > 0$. Therefore,

$$\begin{aligned} D_1(A) - D_2(A) &\leq \sum_i \theta_i(A_i) \max\left(\phi_i^1 - \phi_i^2, 0\right) + \theta_{i^*}(A_{i^*})\left(\phi_i^1 - \phi_i^2\right) \\ &< \sum_i \theta_i(A_i) \max\left(\phi_i^1 - \phi_i^2, 0\right) \end{aligned}$$

□

A.2 Proof of MSP Analysis theorem

Proof. **of Theorem 1**

1. The computational complexity is straight forward. The MSP algorithm first computes the expected value for each of the samples. For every sample this takes nN_j. Once the expected values were computed, computing each of the $\bar{v}_j$ vector is $2n$ operations.

2. Recall that $D_j = \sum_i \phi_i^j \theta_i$. We can rewrite it as

$$D_j = \sum_{i=1}^K \phi_i^j \mu_i + \sum_{i=1}^K \phi_i^j (\theta_i - \mu_i) = E_j + \sum_{i=1}^K \phi_i^j (\theta_i - \mu_i) \quad .$$

Note that for every i, $(\theta_i - \mu_i)$ has a zero mean and variance bounded by $\sigma_{\max}^2$. Since $\phi_i^j \geq 0$ and $\sum_i \phi_i^j = 1$ then the measure $\sum_{i=1}^K \phi_i^j (\theta_i - \mu_i)$ has zero mean and variance bounded by $\sigma_{\max}^2$. Hence, D_j is a measure with mean E_j and variance bounded by $\sigma_{\max}^2$. Since $\bar{E}_j$ is obtained by averaging N_j instances, we get, from Chebyshev's inequality combined with the union bound that

$$\Pr\left[\left\|E_j - \bar{E}_j\right\| > \epsilon\right] \leq \frac{n\sigma_{\max}^2}{N_j \epsilon^2} \quad .$$

Since there are m estimators, $\bar{E}_1, \ldots, \bar{E}_m$, using the union bound we obtain

$$\Pr\left[\sup_j \left\|E_j - \bar{E}_j\right\| > \epsilon\right] \leq \sum_j \frac{n\sigma_{\max}^2}{N_j \epsilon^2} \quad .$$

3. Recall that $\bar{\mu}_i$ is the projection of μ_i on the space $\bar{V} = \text{span}(\bar{v}_1, \ldots, \bar{v}_{m-1})$. Therefore, $\bar{\mu}_i = \arg\min_{v \in \bar{V}} (\|\mu_i - v\|)$. Since $\mu_i \in \text{span}\{v_1, \ldots, v_{m-1}\}$ then $\mu_i = \sum \alpha_j^i v_j$.

$$\begin{aligned} \|\mu_i - \bar{\mu}_i\| &\leq \left\| \sum \alpha_j v_j - \sum \alpha_j \bar{v}_j \right\| \\ &\leq \sum |\alpha|_j \|v_j - \bar{v}_j\| \ . \end{aligned}$$

Hence, if $A = \max_i \sum \left|\alpha_j^i\right|$ then with probability of at least $1 - \frac{n\sigma_{\max}^2}{\epsilon^2} \sum_j \frac{1}{n_j}$

$$\max_i \|\mu_i - \bar{\mu}_i\| \leq \epsilon A \ .$$

Furthermore,

$$\max_{i,i'} |\|\mu_i - \mu_{i'}\| - \|\bar{\mu}_i - \bar{\mu}_{i'}\|| \leq 2\epsilon A \ .$$

□

It is possible to improve upon the bounds presented here. We can get sharper bounds on the probability of success in several ways:

1. If we assume that the sample space is bounded we can use Bernstein's inequality instead of Chebyshev's inequality
2. If we assume that the covariance matrix is diagonal we can replace the union bounded with better concentration of measure bounds
3. If we assume that the distributions are Gaussians we can use tail bounds on these distribution instead of Chebyshev's inequality

To simplify the presentation, we do not derive the bounds for these specific conditions.

A.3 Proof of Theorem 2

The analysis we use several lemmas. To simplify the notation we define the following assumptions:

Assumption 1. *The gap between Φ^1 and Φ^2 is $g > 0$.*

Assumption 2. *Both Φ^1 and Φ^2 are bounded away from zero by $b > 0$.*

Definition 4. *For the set A, we say that the i'th cluster is γ-*big *if $\theta_i(A) \geq 1 - \gamma$ and we say that it is γ-*small *if $\theta_i(A) \leq \gamma$.*

Assumption 3. *For the set A all clusters are either γ-big or γ-small and there exists at least one γ-big cluster.*

Assumption 4. *The classifier h and the estimator e are such that*

$$\left| \Pr_{x,y \sim \nu} [h(x) \neq y] - e \right| \leq \epsilon$$

and

$$\forall I, \quad \Pr_{x,y \sim \nu} [h(x) \neq y] < \Pr_{x,y \sim \nu} [c_I(x) \neq y] + \epsilon$$

where ν is a measure on $X \times \{\pm 1\}$ such that the measure of $B \subseteq X \times \{\pm 1\}$ is

$$\nu(B) = \frac{1}{2} \left(\frac{D_1(\{x \in A \ : \ (x,1) \in B\})}{D_1(A)} + \frac{D_2(\{x \in A \ : \ (x,-1) \in B\})}{D_2(A)} \right) \ .$$

Using these assumptions we turn to prove the lemmas.

Lemma 2. *Under assumptions 2 and 3, if $I = \left\{ i \ : \ \frac{\phi_i^1}{\sum_{i'} \phi_{i'}^1 \theta_{i'}(A)} > \frac{\phi_i^2}{\sum_{i'} \phi_{i'}^2 \theta_{i'}(A)} \right\}$, there are more than a single γ-big cluster and $\gamma < \min(b/2, \ gb/k+3)$ then*

$$\Pr_{x,y \sim \nu} [c_I(x) \neq y] \leq \frac{1}{2} \left(1 - g(1 - \gamma) + \frac{3K\gamma}{b} \right)$$

where ν is as defined in assumption 4. Moreover, the set I contains at least one γ-big cluster but does not contain all the γ-big clusters.

Proof. Let J be the set of γ big clusters. From the definition of γ we have that

$$\begin{aligned}
\Pr_{x,y\sim\nu}\left[c_I\left(x\right)\neq y\right] &= \frac{1}{2}\left(\sum_{i\notin I}\frac{\phi_i^1\theta_{i'}\left(A\right)}{\sum_{i'}\phi_{i'}^1\theta_{i'}\left(A\right)}+\sum_{i\in I}\frac{\phi_i^2\theta_{i'}\left(A\right)}{\sum_{i'}\phi_{i'}^2\theta_{i'}\left(A\right)}\right)\\
&= \frac{1}{2}\left(1-\sum_{i\in I}\left(\frac{\phi_i^1}{\sum_{i'}\phi_{i'}^1\theta_{i'}\left(A\right)}-\frac{\phi_i^2}{\sum_{i'}\phi_{i'}^2\theta_{i'}\left(A\right)}\right)\theta_{i'}\left(A\right)\right)\\
&\leq \frac{1}{2}\left(1-\sum_{i\in I\cap J}\left(\frac{\phi_i^1}{\sum_{i'}\phi_{i'}^1\theta_{i'}\left(A\right)}-\frac{\phi_i^2}{\sum_{i'}\phi_{i'}^2\theta_{i'}\left(A\right)}\right)\theta_{i'}\left(A\right)\right)\\
&\leq \frac{1}{2}\left(1-\sum_{i\in I\cap J}\left(\frac{\phi_i^1}{\sum_{i'\in J}\phi_{i'}^1+\gamma}-\frac{\phi_i^2}{\sum_{i'\in J}\phi_{i'}^2-\gamma}\right)\theta_{i'}\left(A\right)\right)\quad .
\end{aligned}$$

Due to assumption 2

$$\begin{aligned}
\frac{\phi_i^1}{\sum_{i'\in J}\phi_{i'}^1+\gamma} &= \frac{\phi_i^1}{\sum_{i'\in J}\phi_{i'}^1}-\frac{\gamma\phi_i^i}{\left(\sum_{i'\in J}\phi_{i'}^1\right)\left(\sum_{i'\in J}\phi_{i'}^1+\gamma\right)}\\
&\geq \frac{\phi_i^1}{\sum_{i'\in J}\phi_{i'}^1}-\frac{\gamma}{\sum_{i'\in J}\phi_{i'}^1+\gamma}\\
&\geq \frac{\phi_i^1}{\sum_{i'\in J}\phi_{i'}^1}-\frac{\gamma}{b}\quad . \qquad (1)
\end{aligned}$$

Since $\gamma<{}^{b}\!/_{2}$ we have

$$\begin{aligned}
\frac{\phi_i^2}{\sum_{i'\in J}\phi_{i'}^2-\gamma} &= \frac{\phi_i^2}{\sum_{i'\in J}\phi_{i'}^2}+\frac{\gamma\phi_i^2}{\left(\sum_{i'\in J}\phi_{i'}^2\right)\left(\sum_{i'\in J}\phi_{i'}^2-\gamma\right)}\\
&\leq \frac{\phi_i^2}{\sum_{i'\in J}\phi_{i'}^2}+\frac{\gamma}{\sum_{i'\in J}\phi_{i'}^2-\gamma}\\
&\leq \frac{\phi_i^2}{\sum_{i'\in J}\phi_{i'}^2}+\frac{2\gamma}{b}\quad . \qquad (2)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\Pr_{x,y\sim\nu}\left[c_I\left(x\right)\neq y\right] &\leq \frac{1}{2}\left(1-\sum_{i\in I\cap J}\left(\frac{\phi_i^1}{\sum_{i'\in J}\phi_{i'}^1+\gamma}-\frac{\phi_i^2}{\sum_{i'\in J}\phi_{i'}^2-\gamma}\right)\theta_i\left(A\right)\right)\\
&\leq \frac{1}{2}\left(1-\sum_{i\in I\cap J}\left(\frac{\phi_i^1}{\sum_{i'\in J}\phi_{i'}^1}-\frac{\phi_i^2}{\sum_{i'\in J}\phi_{i'}^2}-\frac{3\gamma}{b}\right)\theta_i\left(A\right)\right)\\
&\leq \frac{1}{2}\left(1-\sum_{i\in I\cap J}\left(\frac{\phi_i^1}{\sum_{i'\in J}\phi_{i'}^1}-\frac{\phi_i^2}{\sum_{i'\in J}\phi_{i'}^2}\right)\theta_i\left(A\right)+\frac{3K\gamma}{b}\right)\\
&\leq \frac{1}{2}\left(1-\sum_{i\in I\cap J}g\theta_i\left(A\right)+{}^{gb}\!/_{k+3}\frac{3K\gamma}{b}\right)\\
&\leq \frac{1}{2}\left(1-g\left(1-\gamma\right)+\frac{3K\gamma}{b}\right)\quad .
\end{aligned}$$

Note that we have used the fact that $I \cap J$ is not empty. Otherwise, note that since $\gamma < 1/2$ and from (1) and (2)

$$\begin{aligned}
0 &= \sum_{i\in I}\left(\frac{\phi_i^1\theta_{i'}(A)}{\sum_{i'}\phi_{i'}^1\theta_{i'}(A)} - \frac{\phi_i^2\theta_{i'}(A)}{\sum_{i'}\phi_{i'}^2\theta_{i'}(A)}\right) - \sum_{i\notin I}\left(\frac{\phi_i^2\theta_{i'}(A)}{\sum_{i'}\phi_{i'}^2\theta_{i'}(A)} - \frac{\phi_i^1\theta_{i'}(A)}{\sum_{i'}\phi_{i'}^1\theta_{i'}(A)}\right) \\
&\leq \gamma\sum_{i\in I}\left(\frac{\phi_i^1}{\sum_{i'}\phi_{i'}^1\theta_{i'}(A)}\right) - (1-\gamma)\sum_{i\in J}\left(\frac{\phi_i^2}{\sum_{i'}\phi_{i'}^2\theta_{i'}(A)} - \frac{\phi_i^1}{\sum_{i'}\phi_{i'}^1\theta_{i'}(A)}\right) \\
&\leq \gamma K\frac{1}{2b(1-\gamma)} - (1-\gamma)\sum_{i\in J}\left(g - \frac{3\gamma}{b}\right) \\
&\leq \frac{\gamma K}{b} - 2(1-\gamma)\left(g - \frac{3\gamma}{b}\right) \\
&\leq \frac{\gamma K}{b} - \left(g - \frac{3\gamma}{b}\right) \\
&= \gamma\frac{K+3}{b} - g \ .
\end{aligned}$$

However, since $\gamma < {}^{gb}/_{k+3}$ we obtain $\gamma\frac{K+3}{b} - g < 0$ which is a contradiction. Therefore, $I \cap J$ is not empty. In the same way we can see that $\bar{I} \cap J$ is not empty as well. □

Lemma 3. *Under assumptions 1, 2, 3 and 4 if* $\gamma < \min({}^{b}/_{2}, {}^{gb}/_{3})$ *then*

$$\forall i,\ \theta_i(A\cap h), \theta_i\left(A\cap\bar{h}\right) \notin \left[\gamma + \frac{2\epsilon}{g - \frac{3\gamma}{b}}, 1 - \gamma - \frac{2\epsilon}{g - \frac{3\gamma}{b}}\right] \ .$$

Moreover, if $I = \left\{i : \frac{\phi_i^1}{\sum_i \phi_i^1\theta_i(A)} > \frac{\phi_i^2}{\sum_i \phi_i^2\theta_i(A)}\right\}$ *then* $\theta_i(A\cap h) \geq 1 - \gamma - \frac{2\epsilon}{g-\frac{3\gamma}{b}}$ *iff* i *is a* γ*-big cluster in* A *and* $i \in I$.

Proof. Recall that for every set B, $D_j(B) = \sum_i \phi_i^j\theta_i(B)$. By using h both as a function and as a subset of X we have

$$\begin{aligned}
\Pr_{x,y\sim\nu}[h(x)\neq y] &= \frac{1}{2}\left(\frac{D^1(A\setminus h)}{D^1(A)} + \frac{D^2(h)}{D^2(A)}\right) \\
&= \frac{1}{2}\left(1 - \left(\frac{D^1(h\cap A)}{D^1(A)} - \frac{D^2(h\cap A)}{D^2(A)}\right)\right) \\
&= \frac{1}{2}\left(1 - \left(\frac{\sum_i \phi_i^1\theta_i(h\cap A)}{\sum_i \phi_i^1\theta_i(A)} - \frac{\sum_i \phi_i^2\theta_i(h\cap A)}{\sum_i \phi_i^2\theta_i(A)}\right)\right) \\
&= \frac{1}{2}\left(1 - \sum_i\left(\frac{\phi_i^1}{\sum_{i'}\phi_{i'}^1\theta_{i'}(A)} - \frac{\phi_i^2}{\sum_{i'}\phi_{i'}^2\theta_{i'}(A)}\right)\theta_i(h\cap A)\right)
\end{aligned}$$

Let $I = \left\{i : \frac{\phi_i^1}{\sum_i \phi_i^1\theta_i(A)} > \frac{\phi_i^2}{\sum_i \phi_i^2\theta_i(A)}\right\}$. It follows that $c_I = \arg\min_h \Pr_{x,y\sim\nu}[h(x)\neq y]$ and hence

$$\Pr_{x,y\sim\nu}[c_I(x)\neq y] \leq \Pr_{x,y\sim\nu}[h(x)\neq y] < \Pr_{x,y\sim\nu}[c_I(x)\neq y] + \epsilon \ .$$

Therefore,

$$\begin{aligned}
\epsilon &> \Pr_{x,y\sim\nu}[h(x)\neq y] - \Pr_{x,y\sim\nu}[c_I(x)\neq y] \\
&= \frac{1}{2}\left(\sum_i\left(\mathbb{I}_{i\in I}\theta_i(A) - \theta_i(h\cap A)\right)\left(\frac{\phi_i^1}{\sum_i \phi_i^1\theta_i(A)} - \frac{\phi_i^2}{\sum_i \phi_i^2\theta_i(A)}\right)\right) \\
&\geq \frac{1}{2}\max_i\left(\mathbb{I}_{i\in I}\theta_i(A) - \theta_i(h\cap A)\right)\left(\frac{\phi_i^1}{\sum_i \phi_i^1\theta_i(A)} - \frac{\phi_i^2}{\sum_i \phi_i^2\theta_i(A)}\right)
\end{aligned}$$

Let $J = \{j : \theta_j(A) > 1-\gamma\}$ then

$$\begin{aligned}\frac{\phi_i^1}{\sum_i \phi_i^1 \theta_i(A)} - \frac{\phi_i^2}{\sum_i \phi_i^2 \theta_i(A)} &\geq \frac{\phi_i^1}{\sum_{j\in J}\phi_j^1 + \sum_{j\in J}\phi_j^1 \gamma} - \frac{\phi_i^2}{\sum_{j\in J}\phi_j^2(1-\gamma)} \\ &\geq \frac{\phi_i^1}{\sum_{j\in J}\phi_j^1 + \gamma} - \frac{\phi_i^2}{\sum_{j\in J}\phi_j^2 - \gamma}\end{aligned}$$

and hence

$$\epsilon \geq \frac{1}{2}\max_i \left(\mathbb{I}_{i\in I}\theta_i(A) - \theta_i(h\cap A)\right)\left(\frac{\phi_i^1}{\sum_{j\in J}\phi_j^1 + \gamma} - \frac{\phi_i^2}{\sum_{j\in J}\phi_j^2 - \gamma}\right)$$

or put otherwise

$$\max_i \left(\mathbb{I}_{i\in I}\theta_i(A) - \theta_i(h\cap A)\right) \leq \frac{2\epsilon}{\min_i\left(\frac{\phi_i^1}{\sum_{j\in J}\phi_j^1+\gamma} - \frac{\phi_i^2}{\sum_{j\in J}\phi_j^2-\gamma}\right)} \quad . \tag{3}$$

Note that since $\gamma < b/2$ from (1) and (2) it follows that

$$\begin{aligned}\min_i\left(\frac{\phi_i^1}{\sum_{j\in J}\phi_j^1+\gamma} - \frac{\phi_i^2}{\sum_{j\in J}\phi_j^2-\gamma}\right) &\geq \min_i\left(\frac{\phi_i^1}{\sum_{j\in J}\phi_j^1} - \frac{\phi_i^2}{\sum_{j\in J}\phi_j^2}\right) - \frac{3\gamma}{b} \\ &\geq g - \frac{3\gamma}{b} \quad .\end{aligned}$$

Note that $g - 3\gamma/b > 0$ since $\gamma < gb/3$. If i is such that $\theta_i(A) < \gamma$ then clearly $\theta_i(h\cap A) < \gamma$. However, if $\theta_i(A) > 1-\gamma$ and $i \in I$ then from (3) we have that

$$\begin{aligned}\theta_i(h\cap A) &\geq \theta_i(A) - \frac{2\epsilon}{g-\frac{3\gamma}{b}} \quad . \\ &\geq 1-\gamma-\frac{2\epsilon}{g-\frac{3\gamma}{b}}\end{aligned}$$

If however, $\theta_i(A) > 1-\gamma$ but $i \notin I$ then we can repeat the same argument for $\bar{h}$ to get

$$\theta_i\left(\bar{h}\cap A\right) \geq 1-\gamma-\frac{2\epsilon}{g-\frac{3\gamma}{b}}$$

and thus

$$\theta_i(h\cap A) \leq \gamma + \frac{2\epsilon}{g-\frac{3\gamma}{b}} \quad .$$

□

Lemma 4. *Under assumptions 2,3 and 4, if there is only a single γ-big cluster then*

$$e \geq \frac{1}{2} - \gamma\left(\frac{1}{\gamma + b(1-\gamma)} + \frac{1}{1-\gamma} + \frac{1}{b+\gamma}\right) - \epsilon$$

Proof. Let i^* be such that θ_{i^*} is the single γ-big cluster. For $j = 1, 2$

$$\begin{aligned}\frac{D_j(C_{i^*}\cap A)}{D_j(A)} &= \frac{\phi_{i^*}^j\theta_{i^*}(A)}{\sum_i \phi_i^j\theta_i(A)} \\ &= \frac{1}{1+\frac{\sum_{i\neq i^*}\phi_i^j\theta_i(A)}{\phi_{i^*}^j\theta_{i^*}(A)}} \\ &\geq \frac{1}{1+\frac{\gamma}{b(1-\gamma)}} \quad .\end{aligned}$$

For any h,

$$\begin{aligned}
\left|\frac{D_j(A\cap h)}{D_j(A)}-\theta_{i^*}(h)\right| &\leq \left|\frac{D_j((A\setminus C_{i^*})\cap h)}{D_j(A)}\right|+\left|\frac{D_j(A\cap C_{i^*}\cap h)}{D_j(A)}-\theta_{i^*}(h)\right| \\
&\leq \frac{\gamma}{\gamma+b(1-\gamma)}+\left|\frac{\phi^j_{i^*}\theta_{i^*}(A\cap h)}{D_j(A)}-\theta_{i^*}(h)\right| \\
&\leq \frac{\gamma}{\gamma+b(1-\gamma)}+\left|\frac{\phi^j_{i^*}\theta_{i^*}(\bar{A})}{D_j(A)}\right|+\theta_{i^*}(h)\left|\frac{\phi^j_{i^*}}{D_j(A)}-1\right| \\
&\leq \frac{\gamma}{\gamma+b(1-\gamma)}+\frac{\gamma}{1-\gamma}+\theta_{i^*}(h)\left(1-\frac{\phi^j_{i^*}}{\phi^j_{i^*}+\gamma}\right) \\
&\leq \frac{\gamma}{\gamma+b(1-\gamma)}+\frac{\gamma}{1-\gamma}+\frac{\gamma}{b+\gamma}\ .
\end{aligned}$$

Therefore,

$$\begin{aligned}
\left|\frac{D_1(A\cap h)}{D_1(A)}-\frac{D_2(A\cap h)}{D_2(A)}\right| &\leq \left|\frac{D_1(A\cap h)}{D_1(A)}-\theta_{i^*}(h)\right|+\left|\frac{D_2(A\cap h)}{D_2(A)}-\theta_{i^*}(h)\right| \\
&\leq 2\gamma\left(\frac{1}{\gamma+b(1-\gamma)}+\frac{1}{1-\gamma}+\frac{1}{b+\gamma}\right)\ .
\end{aligned}$$

Hence

$$\begin{aligned}
e &\geq \Pr_{x,y\sim\nu}[h(x)\neq y]-\epsilon \\
&= \frac{1}{2}-\frac{1}{2}\left|\frac{D_1(A\cap h)}{D_1(A)}-\frac{D_2{}^{gb}/_3(A\cap h)}{D_2(A)}\right|-\epsilon \\
&\geq \frac{1}{2}-\gamma\left(\frac{1}{\gamma+b(1-\gamma)}+\frac{1}{1-\gamma}+\frac{1}{b+\gamma}\right)-\epsilon\ .
\end{aligned}$$

□

Lemma 5. *Under assumptions 2,3 and 4, if there are $t>1$ clusters which are γ-big for $\gamma<\min({}^{b}/_{2},\ {}^{gb}/_{k+3})$ then*

$$e\leq\frac{1}{2}\left(1-g(1-\gamma)+\frac{3K\gamma}{b}\right)+2\epsilon$$

and the split induced by h will have at least one $\gamma+\frac{2\epsilon}{g-\frac{3\gamma}{b}}$ big cluster in each side of the split.

Proof. Let $I=\left\{i:\ \frac{\phi^1_i}{\sum_i\phi^1_i\theta_i(A)}>\frac{\phi^2_i}{\sum_i\phi^2_i\theta_i(A)}\right\}$ then from Lemma 2

$$\Pr_{x,y\sim\nu}[c_I(x)\neq y]\leq\frac{1}{2}\left(1-g(1-\gamma)+\frac{3K\gamma}{b}\right)$$

and I contains at least one γ-big cluster but does not contain all the γ big clusters. From Assumption 4 it follows that

$$e\leq\frac{1}{2}\left(1-g(1-\gamma)+\frac{3K\gamma}{b}\right)+2\epsilon\ .$$

Moreover, from Lemma 3 a cluster is $\gamma+\frac{2\epsilon}{g-\frac{3\gamma}{b}}$-big if and only if it is γ-big in A and it is in I. Since I contains a γ-big cluster than $A\cap h$ contains a $\gamma-\frac{2\epsilon}{g-\frac{3\gamma}{b}}$-cluster. However, since I does not contain all the γ-big clusters, there is a $\gamma+\frac{2\epsilon}{g-\frac{3\gamma}{b}}$-cluster in $A\cap\bar{h}$ as well.

□

We are now ready to prove the main theorem.

Proof. **of Theorem 2**

Let $\epsilon < \min\left(\frac{g^2 b}{160K}, \frac{bg}{8K}, \frac{g^2 b}{4K(K+3)}, \frac{\epsilon^* g}{4K}\right)$ and let $\hat{\gamma}_l = \frac{4\epsilon l}{g}$ then for every $l \leq K$ we have

$$\frac{3\hat{\gamma}_l}{b} < \frac{g}{2} \ , \ \hat{\gamma}_l < \min\left(\frac{b}{2}, \frac{gb}{K+3}\right) \ .$$

Let $\delta = \frac{\delta^*}{4K}$ and N_1 be the size of the samples needed for L to return an ϵ, δ good hypothesis. Let

$$N = \max\left(\frac{4N_1}{b}, \frac{2}{b^2}\log\frac{1}{\delta}\right) \ .$$

Note the following, if A is a set that contains at least one γ-big cluster, for $\gamma \leq \hat{\gamma}_K$ then with probability $1-\delta$, a sample of N points from γ contains atleast N_1 points from A. To see that, note that

$$\gamma \leq \frac{4\epsilon K}{g} < \frac{4K}{g} \cdot \frac{gb}{8K} \leq \frac{gb}{2} \leq \frac{1}{2} \ .$$

Since each cluster is bounded away from zero by $b > 0$, the expected number points in A out of a sample of size N is at least $(1-\gamma)\, bN \geq {}^{bN}/_{2}$. From Hoeffding's inequality, we have that for $N \geq \frac{2}{b^2}\log\frac{1}{\delta}$ with a probability of $1-\delta$ there will be at least ${}^{bN}/_{4}$ points in A. Since $N \geq {}^{4N_1}/_{b}$, we conclude that with a probability of at least $1-\delta$, there will be atleast N_1 points from A in a random set of N points. Therefore, with probability $1 - {}^{\delta}/_{2}$, in the first $2K$ calls for the learning algorithm L, for which there was at least one γ-big cluster in the leaf, there were at least N_1 points to train the algorithm from, provided that $\gamma \leq \hat{\gamma}_K$. Hence, with probability $1-\delta$, Assumption 4 holds for the first $2K$ call to the learning algorithm, provided that $\gamma \leq \hat{\gamma}_K$.

Next we will show that as long as Assumption 4 holds, the DSC algorithm will make at most $2K$ calls to L and all leafs will have at least one γ-big cluster for $\gamma \leq \hat{\gamma}_K$. We prove that by induction on the depth of the generated tree. Initially, we have a single leaf with all the points and hence all the clusters are 0-big hence the assumption clearly works. From Lemma 3 it follows that if all the clusters were γ-big or γ-small, and indeed Assumption 4 holds, then the new leafs that are generated by splitting on h will have only γ'-big or γ'-small clusters for $\gamma' \leq \gamma + \frac{2\epsilon}{g - \frac{3\gamma}{b}}$. Note that if $\gamma < {}^{gb}/_{6}$ then $g - {}^{3\gamma}/_{b} > {}^{g}/_{2}$ hence $\gamma' \leq \gamma + {}^{4\epsilon}/_{g}$.

From Lemma 4 and Lemma 5 it follows that every time the DSC algorithm calls the L algorithm, assuming that Assumption 4 holds for this call, one of two things happen, either the algorithm finds a non-trivial split of the clusters, which happens whenever there is more than a single big cluster, in which case $e \leq \frac{1}{2}\left(1 - g\,(1-\gamma) + \frac{3K\gamma}{b}\right) + 2\epsilon$ or otherwise, if there is only a single big cluster, $e \geq \frac{1}{2} - \gamma\left(\frac{1}{\gamma + b(1-\gamma)} + \frac{1}{1-\gamma} + \frac{1}{b+\gamma}\right) - \epsilon$. Note that if $\gamma \leq \min\left({}^{1}/_{2}\,,\, {}^{4K\epsilon}/_{g}\right)$ then

$$\begin{aligned}
\frac{1}{2} - \gamma\left(\frac{1}{\gamma + b\,(1-\gamma)} + \frac{1}{1-\gamma} + \frac{1}{b+\gamma}\right) - \epsilon \ &\geq \ \frac{1}{2} - \frac{4K\epsilon}{g}\left(\frac{2}{b} + 2 + \frac{1}{b}\right) - \epsilon \\
&\geq \ \frac{1}{2} - \frac{K}{gb}\left(12 + 8b + \frac{gb}{K}\right)\epsilon \\
&> \ \frac{1}{2} - \frac{20K}{gb}\epsilon \ .
\end{aligned}$$

Since, $\epsilon < {}^{g^2 b}/_{160K}$, if there was only a single cluster and Assumption 4 holds then $e > \frac{1}{2} - \frac{g}{8}$. However, if there were multiple big clusters then

$$e \leq \frac{1}{2}\left(1 - g\,(1-\gamma) + \frac{3K\gamma}{b}\right) + 2\epsilon \leq \frac{1}{2} - \frac{g}{4} \ .$$

Hence, assuming that Assumption 4 holds for all splits then the algorithm will split every leaf that contain multiple big clusters and will not split any leaf that contain a single big cluster. Therefore, after $K-1$ splits, all the leafs will have a single big cluster. For each of the K leaf, the DSC algorithm will call L once to determine that it contains a single cluster and hence the number of calls to L will be at most $2K-1$ and in each call all the clusters are either γ-big or γ-small for $\gamma \leq \gamma_K = {}^{4K\epsilon}/_{g}$. And therefore, with probability $1-\delta^*$, the DSC algorithm will return a γ_K clustering tree. Since

$$\gamma_K = \frac{4\epsilon K}{g} \leq \frac{4K\frac{\epsilon^* g}{4K}}{g} = \epsilon^* \ ,$$

this is an ϵ^* clustering tree. □