An Investigation of Computational and Informational Limits in Gaussian Mixture Clustering

Nathan Srebro[†] NATI@CS.TORONTO.EDU
Gregory Shakhnarovich[‡] GREGORY@CS.BROWN.EDU
Sam Roweis[†] ROWEIS@CS.TORONTO.EDU
[†] Dept. of Computer Science, University of Toronto, Toronto, Ontario, CANADA
[‡] Dept. of Computer Science, Brown University, Providence, Rhode Island, USA

Abstract

We investigate under what conditions clustering by learning a mixture of spherical Gaussians is (a) computationally tractable; and (b) statistically possible. We show that using principal component projection greatly aids in recovering the clustering using EM; present empirical evidence that even using such a projection, there is still a large gap between the number of samples needed to recover the clustering using EM, and the number of samples needed without computational restrictions; and characterize the regime in which such a gap exists.

1. Introduction

Consider clustering a collection of points by fitting a mixture-of-Gaussians model to the data. Viewed as a problem of optimizing an objective function, such as the likelihood, this problem seems to be hard in the traditional worst-case sense. On the other hand, when the data is inherently clustered, and enough data is available, local search methods typically succeed in optimizing the objective and recovering the clustering. This leads to the conventional wisdom that *"clustering is not hard—it is either easy, or not interesting"*. How true is this statement? Is there a regime in which clustering is hard even though it is interesting? When *is* clustering hard?

Lately, a series of theoretical results established that if data is generated from an adequately separated mixture of Gaussians, and enough samples are available, then clustering is in fact easy—polynomial time algorithms exist that can recover (almost exactly) the correct clustering (see Section 2.1). These results provide an upper bound on the **computational limit** for clustering—the minimum number of samples and minimum separation between clusters required to tractably recover the clustering.

At the other extreme, when too few samples are available, the correct model cannot be recovered, simply because there is not enough information in the data. Ignoring computational issues, one can ask: How much information is necessary in order to recover the clustering by any procedure? Focusing on the likelihood, how many samples are necessary for the maximum likelihood model to resemble the correct clustering with high probability, i.e. what is the **informational limit** for clustering?

We would like to study the relationship between these computational and informational requirements. For example, is learning a Gaussian mixture always computationally easy when it is statistically possible? Or is there a gap between the computational limit and the informational limit, i.e. a regime in which clustering is hard, even though the optimal (maximum likelihood) clustering is statistically meaningful? If so, can one quantify the *excess information* needed for computational tractability?

In this paper, we investigate these questions through massive simulations on randomly generated data sets. Motivated by the theoretical results mentioned above, we would like to understand (1) whether these results can suggest useful practical methods or modifications to the popular EM algorithm; (2) what is the actual computational limit, considering the results only provide upper bounds; (3) whether there is still a gap between this computational limit and the statistical limit. More broadly, we would like to understand the behavior of the likelihood function, of maximum likeli-

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

hood estimation, and of the computational difficulties of searching for it.

We focus on the simplest possible setting—a uniform mixture of equa-variance spherical Gaussians arranged symmetrically on the vertexes of a simplex. We show that projecting the data onto the leading principal components, as suggested by the theoretical results, can greatly aid in recovering the clustering, also when using EM. We then present evidence that there is still a regime in which EM, even using such a projection, fails to recover the clustering, while the maximum likelihood model succeeds. That is, a regime in which clustering is "interesting" but "hard". Based on our empirical simulations, we also derive a quantitative characterization of this regime.

2. Background

Most formulations of clustering are either known to be,or suspected to be hard in the worst-case. That is, the runtime of algorithms that, for *any* data set, *always* find clusters minimizing some objective, must grow drastically with the amount of available data. However, our interest here is not with worst-case behavior on *any* data set, but only in recovering the clustering from data that does actually have some cluster structure—in our case, data that is generated from a uniform mixture of k equa-variance spherical Gaussians, where each data point X_t is i.i.d. with density:

$$f(x) \propto \frac{1}{k}\sum_{i=1}^{k}\exp(-\frac{\|x-\mu_i\|^2}{2\sigma^2}) \qquad (1)$$

More data is a blessing that can aid in making computation easier, not a curse as in traditional worst-case analysis. We survey several results establishing that the problem is, in fact, easy when data is abundant. Although most of the results apply to more general settings, we describe them in the context to data generated as in equation (1), where (without loss of generality) $\sigma^2 = 1$, and the minimum separation between centers is s, i.e. $\|\mu_i - \mu_j\| \geq s$ for any $i \neq j$.

2.1. Proper Learning

When data is generated from a mixture of well-separated Gaussians, the modes of the mixture are close to its centers. Unfortunately, in high dimensions, a very large sample is required in order to identify the modes. Dasgupta (1999) suggested projecting the sample to a random subspace of dimension $\Theta(\log k)$, and showed that if the separation between Gaussians is $s > \frac{1}{2}d^{1/2}$, then the modes of the distribution in this subspace still correspond to the centers, and can be identified, with probability $1-\delta$, from a sample of size $k^{\Omega(\log^2 1/\delta)}$. Arora and Kannan (2001) later improved the minimum required separation to $s = \Omega(d^{1/4}\log(d))$, using either random projections, or a method based on the fact that with this separation, distances between points in the same cluster are lower then distances between points in different clusters.

When the separation is less than $d^{1/4}$, distances between points are no longer enough in order to separate between the clusters. Vempala and Wang (2004) show that projecting the data to its first k principal components (as in PCA), instead of using a random projection, allows identification of much less separated Gaussians. They show that with a separation of $s = \Omega(k^{1/4}\log^{1/4} dk)$ and a sample of size of $\Omega(d^3k^2\log dks)$, a k-dimensional principal component (PC) projection of the data preserves enough separation between centers of spherical Gaussians such that after such a projection, the Gaussians can be identified by methods similar to those discussed above. These techniques have recently been extended also to non-spherical Gaussians, but the separation required is somewhat higher (Kannan et al., 2005; Achlioptas & McSherry, 2005).

The main thrust of the above results is providing conditions under which the Gaussians are well-separated and easily identifiable (perhaps after a projection), such that even the simplest algorithms can recover them. In general, the results depend on all (or most) distances (after the projection) between points in the same cluster being smaller then distances between points in different clusters. In such extreme cases, local search methods such as EM can also easily recover the clustering. In fact, Dasgupta and Schulman (2000) showed that with a separation of $\Omega(d^{1/4})$ and a number of samples polynomial in k, two rounds of EM are enough in order to get fairly close to the correct centers. This is provided that instead of searching over models with k centers, the first round of EM uses $\Theta(k\log k)$ centers, and those are then pruned down to k far-away, but well used, centers.

The precise distance-based or mode-based methods suggested by the above results should therefore not be regarded as practical alternatives to local search heuristics such as EM, but rather as theoretically analyzable methods. These methods also often involve many parameters that need to be carefully selected, and the theory does not provide for an optimal choice of the parameters for finite sample sizes. In any case, Dasgupta and Schulman's result suggests that, perhaps after a PC projection, we might as well use EM, first allowing for $O(k\log k)$ centers and then pruning to k.

Reference	Separation	Sample Complexity	
Kumar et al. (2004)	$\Omega(d^{\frac{1}{2}}k^{\frac{1}{2}})$		(best possible using the approx. guarantee)
Dasgupta (1999)	$s > \frac{1}{2}d^{\frac{1}{2}}$	$d \cdot \mathrm{poly}(k)$	Random projection, then mode finding
Dasgupta and Schulman (2000)	$s > \Omega\left(d^{\frac{1}{4}}\right)$	$d \cdot \mathrm{poly}(k)$	2 rounds of EM with pruning
Arora and Kannan (2001)	$\Omega\left(d^{\frac{1}{4}}\log d\right)$	large	Inter-cluster distances $\gg$ intra-cluster dist
Vempala and Wang (2004)	$\Omega\left(k^{\frac{1}{4}}\log dk\right)$	$\Omega\left(d^3k^2\log(dk/s)\right)$	After PCA: inter-clust $\gg$ intra-clust dist
Kannan et al. (2005)	$\Omega\left(k^{\frac{5}{2}}\log kd\right)$	$\Omega\left(k^2d\log^5(d)\right)$	Iterative PCA and distance-based
Achlioptas and McSherry (2005)	$s > 4k + o(k)$	$\Omega(k^2d)$	Iterative PCA and distance-based

Table 1. Proper learning results applied to a uniform mixtures of equa-variance spherical Gaussians

2.2. Approximation Algorithms

A separate line of research concerns polynomial time guaranteed approximation algorithms for various clustering objectives. We consider here the current state-of-the-art approximation algorithm for the k-means objective and study whether such an approximation can be useful for clustering data generated from a high-dimensional Gaussian mixture.

The k-means objective for a model consisting of k centers $\mu_1, \ldots, \mu_k$ is given by:

$$\sum_{t=1}^{n} \min_{i=1..k} \|x_t - \mu_i\|^2 \tag{2}$$

This is the negative log-likelihood of the mixture (1), as $\sigma \to 0$. Kumar et al. (2004) present an algorithm that, for any input data set of n points in $\mathbb{R}^d$ and target precision ϵ, is guaranteed to find a model for which the k-means objective is within a multiplicative factor of $(1+\epsilon)$ of optimal, in time $O(2^{(k/\epsilon)^{\mathrm{const}}}dn)$.

To understand if this approximation algorithm can be used to recover a well-separated mixture of Gaussians in high dimensions, consider a uniform mixture of two unit-variance Gaussians in $\mathbb{R}^d$ with centers at $\mu_1 = (-s/2, 0, 0, 0, \ldots)$ and $\mu_2 = (s/2, 0, 0, 0, \ldots)$. As more points are generated from this mixture, the optimal 2-means solution for the sampled points approaches the true centers μ_1 and μ_2, and the k-means objective value of this solution is tightly concentrated around $\mathbf{E}\left[\sum_{i=t}^{n} \min_{i=1} \|X_t - \mu_i\|^2\right] \approx dn$. Consider the alternative trivial solution $\tilde{\mu}_1 = \tilde{\mu}_2 = 0$ (i.e. both centers at the origin). The k-means objective value of this solution is tightly concentrated around $\mathbf{E}\left[\sum_{t=1}^{n} \|X_t\|^2\right] = ((s/2)^2 + d)n$. Therefore, for $\epsilon > s^2/(4d)$, the trivial solution is a valid $(1+\epsilon)$ approximation to the optimal solution. To preclude this possibility, we must insist on $\epsilon < s^2/(4d)$, resulting in a run-time guarantee of $O\left(2^{(kd/s^2)^{\mathrm{const}}}dn\right)$. This suggests that the approximation algorithm might be useful for recovering a mixture in polynomial time only if $s = \Omega(\sqrt{kd})$.

3. Fitting a Gaussian Mixture with EM

We generated data from uniform mixtures of unit-variance spherical Gaussians (equation (1), with $\sigma^2 = 1$), centered at the vertexes of a simplex. That is, the distance between every two centers is exactly s. We then attempted to estimate the centers using the EM method. We fix the covariance matrices to the true (identity) covariance matrices and the mixing proportions to the true (uniform) proportions, and estimate only the centers. The EM updates are then given by

$$\begin{aligned} \textbf{E step:} \quad & \hat{p}_{it} \propto \exp\left(-\tfrac{1}{2}\|\hat{\mu}_i - x_t\|^2\right) \\ \textbf{M step:} \quad & \hat{\mu}_i = \sum_t \hat{p}_{it}x_t / \sum_t \hat{p}_{it} \end{aligned} \tag{3}$$

where $\hat{p}_{it}$ is the estimated posterior probability of point t being assigned to center i. We initialize the centers $\hat{\mu}$ to a random subset of points from the sample, and iterate (3) to convergence. We repeat this procedure several times, each time initializing the centers to a different random subset of points, and select the model with the highest (training) likelihood.

3.1. Dimensionality Reduction

Following the ideas put forth by Vempala and Wang (2004), we also experimented with reducing the dimensionality of the data. We projected the data to its first $k-1$ PCs and ran EM until convergence on the resulting k–1 dimensional data. We then used the estimated posteriors $\hat{p}_{it}$ to estimate centers in the original d-dimensional space, and continued running EM in the original d-dimensional space to convergence.

In order to understand the validity of this approach, consider the covariance matrix of a mixture of spherical Gaussians given by (1). When generating a point X, consider the choice of center as a random vector M, taking one of k values $\mu_1, \ldots, \mu_k$, and such that $X|M \sim \mathcal{N}(M, \sigma^2 I)$. We then have $\mathbf{E}[X] = \mathbf{E}[M]$ and $\mathrm{Cov}[X] = \mathrm{Cov}[M] + \sigma^2 I$, with $\mathrm{Cov}[M]$ of rank at most $k-1$. The $k-1$ principal directions of variation of X about its mean therefore span the centers. In the large

sample limit, the empirical $(k-1)$-dimensional PC subspace of the data converges to the true $k-1$ PC subspace of the distribution. With an infinite number of samples, we therefore reduce the problem to clustering in a $k-1$ dimensional space, without reducing the separation between the centers. With a finite number of samples, we only approximate this subspace spanned by the centers, and do lose some separation. The less samples we have, the more separation we lose.

3.2. Pruning

We also experimented with running EM using more than k centers, as suggested by Dasgupta and Schulman (2000). We ran EM until convergence with $k\log(k)$ centers, estimating the mixing proportions rather than fixing them as we did previously. We then pruned down to only k centers using the method suggested by Dasgupta and Schulman (2000), and continued EM again until convergence, this time fixing the mixing proportions to the true, uniform, proportions.

The rational for such an approach is that if one of the k components is not represented among the k random initial centers, we might completely miss this component and use multiple centers to explain a single other component. Initializing to $k\log(k)$ random centers ensures us that, with high probability, all components will be represented.

Combined with dimensionality reduction, our procedure first projects the data onto its first $k-1$ PCs and then runs EM with $k\log(k)$ centers in the PC subspace. We then prune down to k centers and continue running EM until convergence with k centers in the PC subspace. Finally, we return back to the original high dimensional space and run EM until convergence.

3.3. Empirical Comparison

We experimented with a wide range of dimensionalities d, number of clusters k, separations s and sample sizes n. Figure 1 demonstrates the quality of the solution found by the different EM variants for a specific setting of k,d and s and is typical for most high-dimensional scenarios. When data is not highly abundant, the PC projection indeed helps in finding a higher likelihood solution, reducing by more than a factor of two the sample size required for achieving a low clustering error. In this and all other experiments conducted, the PC projection never hurts, and often helps significantly at finding higher likelihood and lower error models.

The effect of pruning was less dramatic, especially in conjunction with a PC projection. Combined with a PC projection, pruning did increase the likelihood and reduce the clustering error in significantly more experiments than experiments in which it had a reverse effect. However, when the true centers were on a simplex with a inter-center separation of $s < 10$, the differences in likelihood and clustering error between using only a PC projection and using a PC projection and pruning were always extremely small. It appears that at least under such conditions, the true centers can be learned even if we do not initially choose one random center per component[1].

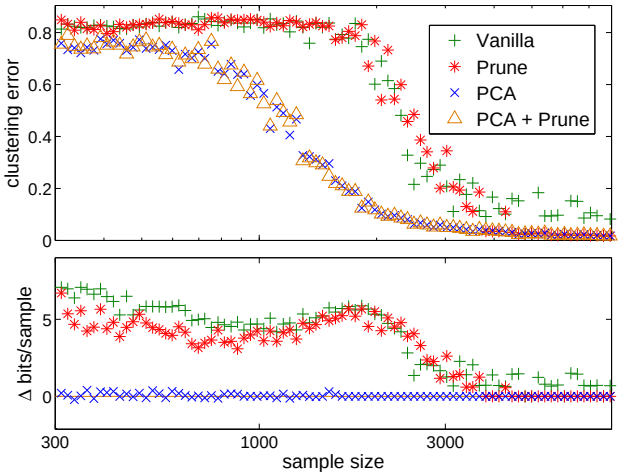

Figure 1. Comparison of EM variants for $k = 16, d = 1024, s = 6$. Top panel: clustering error (fraction of points whose closest estimated center does not correspond to their closest true center, under the optimal correspondence between true and estimated centers). Bottom panel: difference of log-likelihood between each variant and PCA+Prune. 'Vanilla' is regular EM; 'Prune' starts with $k\ln k$ centers; 'PCA' uses a PC projection. For each method, the highest likelihood of ten repetitions was used.

4. Behavior of Learned & ML Models

An investigation of the clustering error, as in the top panel of Figure 1, can reveal how well EM succeeds in recovering the true clustering. In those cases in which the clustering is not fully recovered, we would like to understand if this is because EM failed to find the global ML model, or because there is not enough information in the sample (and thus even the global

[1]The effects of pruning were more noticeable for very large separations, e.g. $s = 15, 20$. We briefly describe the results for these experiments: In such separations when initializing to only k random centers, components were often left out and never discovered, while initializing to $k\log k$ centers alleviated the problem. The effect was also strong when the centers were arranged along a line, rather then on a simplex, in which case even with a separation of 2 between consecutive centers, and eight centers in all, pruning was required in order to find all centers.

ML estimate is far from the true model).

One indication for whether EM fails or succeeds in recovering the ML model is the number of EM repetitions with random initializations that lead to the same highest-likelihood model. If using EM starting from randomly selected centers can find the maximum likelihood solution with reasonable probability, we would expect multiple repetitions of EM to lead to the same (maximum likelihood) model. Therefore, if each of many repetitions of EM leads to a different solution, we can conclude that EM's success probability is low. However, even if using such a test enables us to conclude that we are not finding the ML solution, we can not know if that ML solution is actually closer to the true model.

4.1. The Local Maximum Likelihood Estimate

In order to attempt to answer this last question, and also better decide if EM is finding the maximum likelihood solution, we can "cheat" and run EM until convergence, *starting from centers initialized to the true centers used to generate the data* (this would of course be impossible when confronting real data). The idea is that if the ML solution is in fact close to the true model, then a local search using EM will be able to get to it starting from the true model. This method is by no means guaranteed to find the true ML solution, and in fact often finds solutions with lower likelihood then those found with EM from a random initialization. What we find is the "peak" (local maximum) of the likelihood nearest the true model.

As we see in the top panel of Figure 2, the behavior of this "peak" (plotted as "InitTrue") is not monotonic. For very large samples the peak is very close to the true model, leading to very low clustering error. As the sample size decreases, the "InitTrue" solution first becomes worse, but then when very few samples are available, its clustering error actually improves. A possible explanation for this is that as the sample size decreases, two things happen to the likelihood function. One effect is that as the empirical distribution drifts away from the true generating distribution, so does the shape of the likelihood surface change, and the peak which, at the infinite sample limit is on the true model, gradually drifts away from it. The other effect is that as the sample size decreases, the likelihood surface becomes more jagged and new peaks (local maxima) are introduced, including peaks near the true model. Starting a local search at the true model can end up in a newly formed peak very close to it.

In order to try to verify the above explanation and track the peak as it drifts away from the true model,

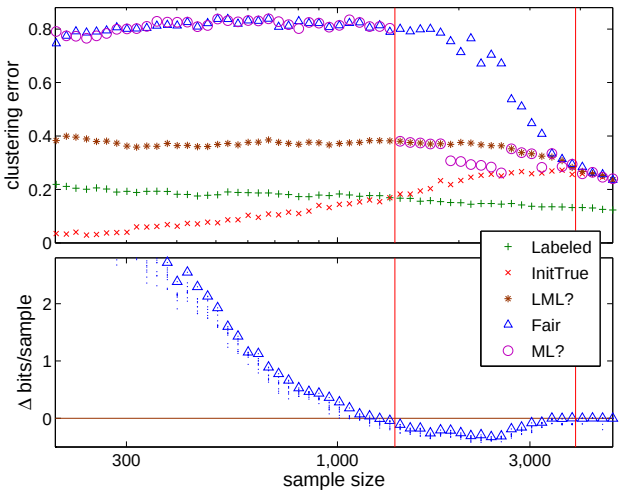

Figure 2. Behavior of different reconstructions for $k = 16$, $d = 512$, $s = 4$. Top panel: clustering error (see Figure 1). Bottom panel: difference in log-likelihood between 'Fair' and 'LML?'. '**Fair**' is the highest likelihood of 10 repetitions of EM, starting with $k \ln k$ centers, and using a PC projection. In the bottom panel, each dot is one of the 10 run; '**InitTrue**' is EM starting from the true centers; '**LML?**' is EM starting from the 'LML?' model with more samples; '**Labeled**' is centers estimated using the true labels; '**ML?**' is highest likelihood of all of the above, as well as EM starting with 'ML?' models with more samples. Vertical lines delineate between the three phases.

we conducted the following experiment: starting with a very large data set, we gradually removed samples from the data set, and after each removal, ran EM (with no PC projection) to convergence starting with the solution we found at the previous iteration (i.e. with a slightly larger data set). The hope is that we start on the true model (or a model very close to it) and continue tracking this peak without ever jumping to a different (perhaps higher) mode of the likelihood. The model found using this procedure is labeled "LML?" in Figure 2. This model is identical to "InitTrue" for large sample size. However, for lower sample sizes the models are different: the likelihood of the "InitTrue" model is worse than that of the "LML?", but its clustering error is better. This probably reflects a lower peak of the likelihood function, closer to the true model. We suspect that our "LML?" model is the local maximum likelihood model discussed by Redner and Walker (1984): a maximum likelihood model in some small neighborhood of the true model, whose deviation from the true model is given by the Fisher information.

In the large sample limit, the LML model, which is the true model, is also the global maximum likelihood model. However, as the sample size decreases, the random variations of the likelihood increase. Eventually,

the random variations cause the likelihood at some unrelated model to be higher than the likelihood of the LML model and the global ML model is no longer related to the true model. At this point, the clustering is essentially un-reconstructible, at least using maximum likelihood estimation.

4.2. Clustering can be Interesting, but Hard

We are now ready to analyze the behavior of our "fair" reconstruction, learned without cheating, using EM from a random starting location, with a PC projection and pruning, relative to the suspected LML and other candidate ML models attained by "cheating" and knowing the true generating distribution.

In the example of Figure 2, for sample sizes beyond ≈4000, the fact that EM always leads to the same model, and comparison with the other "cheating" models, suggests that fair EM succeeds in learning the global ML model, and this model corresponds reasonably well to the true model. With less than ≈1400 samples, the model learned fairly is *not* the maximum likelihood model, but it does have higher likelihood than the LML model, and we cannot find any model with higher likelihood and consistently lower clustering error. This suggest that in this regime the clustering is essentially un-reconstructible. However, for sample sizes between 1400 and 4000, not only does (fair) EM fail at finding the global ML model, but a higher likelihood model (namely, the LML model) is much better at capturing the true clustering. It seems that in this regime the global ML model is hard to find *and* interesting—this represents a gap between the informational and computational requirements for clustering.

From the results presented in Figure 2, and similar results for other parameter settings, we can identify three distinct possibilities for the behavior of the model learned by EM, with PC projection and pruning, versus the maximum likelihood model:

Random Phase EM fails to find the ML model, but the ML model does not correspond to the correct clustering any better then the model found by EM.

Gap Phase EM fails to find the ML model, which *does* correspond better to the correct clustering.

Success Phase EM succeeds in finding the ML model.

The Gap phase represents the regime in which there is a difference between how well the clustering can be reconstructed with unlimited computational resources, and how well is can be reconstructed using EM. Of particular interest is the extent of the Gap, which we measure through the ratio $n_{\mathrm{G}\to\mathrm{S}}/n_{\mathrm{R}\to\mathrm{G}}$, where $n_{\mathrm{R}\to\mathrm{G}}$ and $n_{\mathrm{G}\to\mathrm{S}}$ are the sample sizes at which the phase transitions occur ($n_{\mathrm{R}\to\mathrm{G}} = 1388$ and $n_{\mathrm{G}\to\mathrm{S}} = 3930$ in the example of Figure 2). Suppose we would like to use ML estimation in order to find a model with clustering error below some target level. Even though a sample of size $n_{\mathrm{R}\to\mathrm{G}}$ might be enough for the ML model to achieve this goal, we might need many more samples, perhaps as many as $n_{\mathrm{G}\to\mathrm{S}}$, in order to ensure that the model found by EM (with a PC projection) achieves the desired goal. The ratio $n_{\mathrm{G}\to\mathrm{S}}/n_{\mathrm{R}\to\mathrm{G}}$ therefore bounds the factor by which we might need to increase the sample size in order to obtain computational tractability.

In the next Section, we analyze the quantitative behavior of the phase transitions as a function of the dimensionality, number of clusters and inter-cluster separation, and pay particular attention to the extent of the Gap phase.

5. Analysis of the Phase Transitions

Using methods described in the previous Section, we estimated the location of the "Random", "Gap" and "Success" phase transitions for different numbers of clusters, dimensionality and separations. The results for some values are presented in Figure 3. The sample sizes $n_{\mathrm{R}\to\mathrm{G}}$ and $n_{\mathrm{G}\to\mathrm{S}}$ at the phase transitions display a clear affine dependence on the dimensionality d and on the number of clusters k.

Analyzing also the dependence on the separation, we obtain the following monomial[2] models for the phase transitions (see Figure 4):

$$n_{\mathrm{R}\to\mathrm{G}} = 131 \cdot k \cdot d / s^{4.8} \qquad n_{\mathrm{G}\to\mathrm{S}} = 9.7 \cdot k \cdot d / s^{2.2} \quad (4)$$

It is interesting to observe how the extent of the "Gap" phase changes. Although the ratio $n_{\mathrm{G}\to\mathrm{S}}/n_{\mathrm{R}\to\mathrm{G}}$ is roughly independent of the dimensionality and number of clusters, it increases super-quadratically with s. The Gap phase, a regime in which finding the best statistically possible clustering is hard, is much more pronounced when the separation is large. The model also predicts that for $s < 2.8$, the Gap phase will vanish. This prediction is consistent with some preliminary experiments, and we hope larger scale experiments will

[2] We do observe a small, but statistically distinct from zero, intercept (non-homogeneous term) in the dependence of $n_{\mathrm{R}\to\mathrm{G}}$ and $n_{\mathrm{G}\to\mathrm{S}}$ on d, and especially k. However, to reduce the number of parameters fit, we insist on a (homogeneous) linear fit when studying also the dependence on the separation.

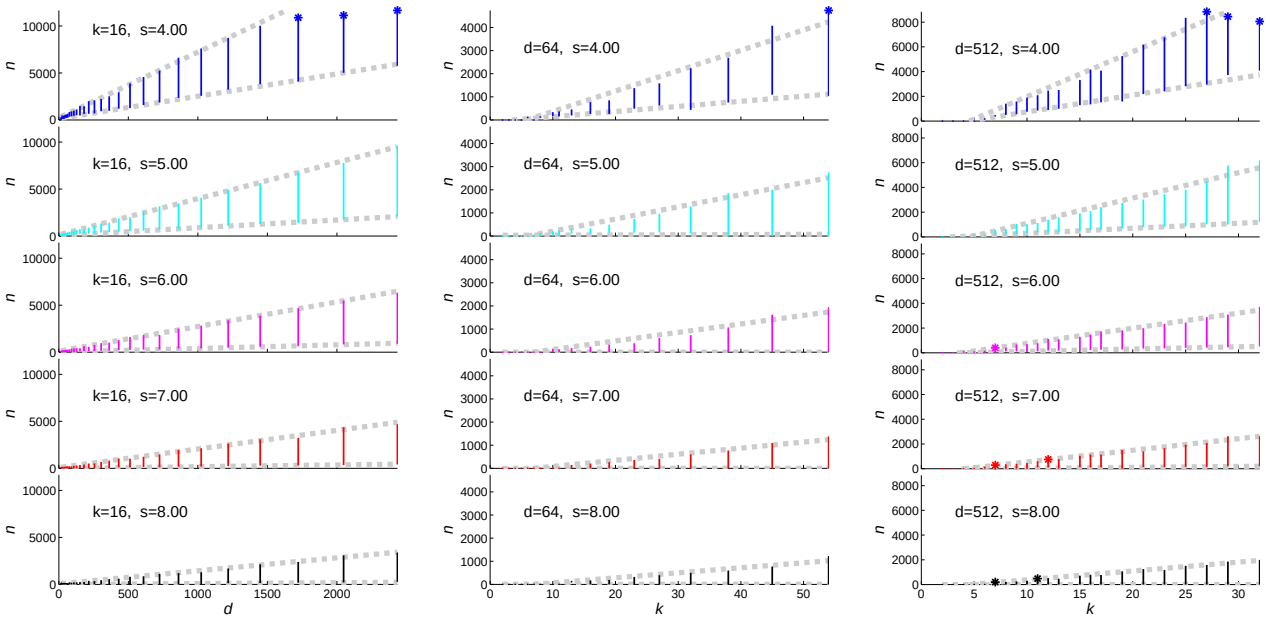

Figure 3. Sample sizes in the "Gap" phase as a function of the number of clusters k, dimensionality d and separation s. Each vertical bar corresponds to an experiment with fixed k, d, s and connects the observed values of the sample sizes $n_{\mathrm{R}\to\mathrm{G}}$ and $n_{\mathrm{G}\to\mathrm{S}}$ at transitions from the "Random" to the "Gap" and from the "Gap" to the "Success" phases. When the largest sample size experimented with was still not in the "Success" phase, an asterisk marks this sample size. For each graph, and each of the phase transitions, a dashed line indicates the best-fit (least-squares) affine model of the form $n = \alpha k + \beta$ or $n = \alpha d + \beta$.

confirm it.

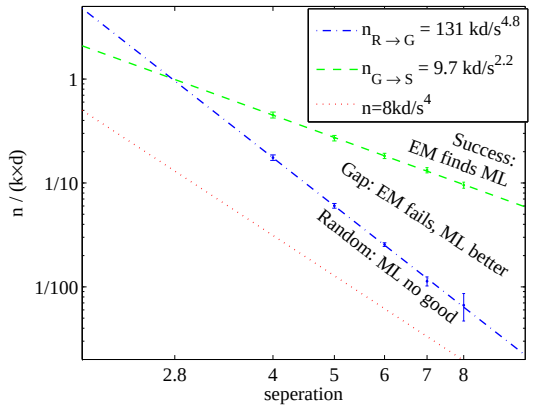

Figure 4. Monomial model for the phases. For each separation s and each of the two phase transitions, the best-fit (least-square) coefficients for the linear models $n = \alpha(s) \cdot k \cdot d$ are plotted, with 95% confidence intervals. The lines (in the log-log plot) are best-fit (least-square) models of the form $n = \alpha k d s^{\beta}$. Dotted line: sample size, for $k = 2$ and $d \to \infty$, where ML starts being correlated with the true centers (Watkins & Nadal, 1994).

6. Discussion

We presented an empirical investigation of the interplay between computational and informational limits in clustering, and a simple quantitative model for the regime in which there is a gap between them. According to our findings, Gaussian mixture clustering is "easy" with a sample size depending only linearly on the number of clusters and the dimensionality, as opposed to the higher order dependencies required by the theoretical guarantees (Table 1). Our results suggest that the gap between the statistical and computational limits narrows, and perhaps even disappears, as the separation decreases. This contrasts with the theoretical guarantees, which only hold for very large separations. It is also interesting to compare the dependence on the separation to the dependence in other studies of the informational limit: An analysis of Gaussian mixture clustering with two clusters and in the limit of infinite dimensionality (Watkins & Nadal, 1994; Barkai & Sompolinksy, 1994, and others), indicated that the sample size required for ML estimation to start correlating with the true clustering is $n = 8d/s^4$ (plotted in Figure 4).

Despite the strong evidence for identifying the three phases, we must qualify our results as we cannot be sure of finding the true maximum likelihood model. In particular, in what we describe as the "Gap Phase", although the EM solutions have lower likelihood than our suspected LML model, we cannot preclude the possibility that the real ML model is hard-to-find and far away from the true model. Similarly, in what we

decide is the "Random Phase", there may well be a hard-to-find ML model, perhaps not as close to the true model as "InitTrue" or even "LML", but closer than the unrelated models EM finds.

Furthermore, our analysis of the computational limit is limited to a specific learning algorithm, and our investigation of the informational limit focuses on maximum-likelihood estimation. It is certainly possible that a better algorithm exists which does efficiently recover the clustering whenever it is statistically recoverable. It is also possible that other estimates work better than the maximum likelihood estimate. In particular, a maximum-likelihood estimate with an overstated variance might sometimes be better (Barkai & Sompolinksy, 1994). Ultimately, we would like to obtain sharp theoretical quantitative evaluations of the computational and informational limits that are independent of a particular algorithm or estimator, and resolve the open question of whether there is indeed a gap in which a clustering is statistically recoverable, but no efficient algorithm can recover it.

Another possible "gap" between the computational and statistical requirements concerns the large sample regime with a small separation. All the theoretical results described in Section 2.1 require, beyond a large enough sample, also a minimal separation between the Gaussians. A mixture of Gaussians that is not well separated might not correspond to a reasonable "clustering", but a ML estimate will still converge to the correct model with enough samples, for any separation. Is some minimum separation indeed required in order for the estimation problem to be tractable, even with a large sample? If so, how does this limit compare to the minimal separation in which the mixture corresponds to a "clustering" in some sense (e.g. the modes of the mixture still correspond to its components, or the component from which points were generated can be identified with reasonable accuracy).

The investigation in this paper is of a scenario in which data is sampled from a symmetric uniform mixture of spherical Gaussians with known, and equal, variances. Even so, there are four parameters to consider (number of clusters, separation, dimensionality and sample size), requiring an unwieldy number of simulations to cover their joint space. It would of course be interesting to understand the possibilities of learning in more general settings, and of the effect of a non-symmetric center configuration (we suspect the symmetric configuration is the hardest). Furthermore it is possible that the "true" generating process does not exactly follow this model, but the data is still separated enough into localized clusters. In such cases it is still possible to recover the clustering by fitting a Gaussian mixture model, and local search methods typically suffice if enough data is available. One can therefore hope to extend the analysis also to such scenarios, characterizing the properties of the true clustering that make it recoverable with a large enough sample.

Acknowledgments We would like to thank Joachim Buhmann and Tali Tishby for pointing us to the relevant literature in the physics community, and Sanjoy Dasgupta for a useful discussion and suggestions.

References

Achlioptas, D., & McSherry, F. (2005). On spectral learning of mixtures of distributions. *18th Annual Conference on Learning Theory (COLT)*.

Arora, S., & Kannan, R. (2001). Learning mixtures of arbitrary gaussians. *Proceedings of the thirty-third annual ACM symposium on Theory of computing*.

Barkai, N., & Sompolinksy, H. (1994). Statistical mechanics of maximum-likelihood desnsity estimation. *Physical Review E*, *50*, 1766–1769.

Dasgupta, S. (1999). Learning mixtures of gaussians. *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*.

Dasgupta, S., & Schulman, L. (2000). A two-round variant of em for gaussian mixtures. *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence*.

Kannan, R., Salmasian, H., & Vempala, S. (2005). The spectral method for general mixture models. *18th Annual Conference on Learning Theory (COLT)*.

Kumar, A., Sabharwal, Y., & Sen, S. (2004). A simple linear time $(1+\epsilon)$-approximation algorithm for k-means clustering in any dimensions. *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*.

Redner, R. A., & Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, *26*, 195–239.

Vempala, S., & Wang, G. (2004). A spectral algorithm for learning mixture models. *J. Comput. Syst. Sci.*, *68*, 841–860.

Watkins, T., & Nadal, J. (1994). Optimal unsupervised learning. *J. Phys. A*, *27*, 1899–1915.