A Rigorous Analysis of Population Stratification with Limited Data.

Kamalika Chaudhuri[*], Eran Halperin[†], Satish Rao[*], Shuheng Zhou[‡]

November 2, 2006

Abstract

Finding the genetic factors of complex diseases such as cancer, currently a major effort of the international community, will potentially lead to better treatment of these diseases. One of the major difficulties in these studies, is the fact that the genetic components of an individual not only depend on the disease, but also on its ethnicity. Therefore, it is crucial to find methods that could reduce the population structure effects on these studies. This can be formalized as a clustering problem, where the individuals are clustered according to their genetic information.

Mathematically, we consider the problem of clustering bit "feature" vectors, where each vector represents the genetic information of an individual. Our model assumes that this bit vector is generated according to a prior probability distribution specified by the individual's membership in a population. We present methods that can cluster the vectors while attempting to optimize the number of features required. The focus of the paper is not on the algorithms, but on showing that optimizing certain objective functions on the data yields the right clustering, under the random generative model. In particular, we prove that some of the previous formulations for clustering are effective.

We consider two different clustering approaches. The first approach forms a graph, and then clusters the data using a connected components algorithm, or a max cut algorithm. The second approach tries to estimate simultanously the feature frequencies in each of the populations, and the classification of vectors into populations. We show that using the first approach $\Theta(\log N/\gamma^2)$ data (i.e., total number of features times number of vectors) is sufficient to find the correct classification, where N is the number of vectors of each population, and γ is the average ℓ_2^2 distance between the feature probability vectors of the two populations. Using the second approach, we show that $O(\log N/\alpha^4)$ data is enough, where α is the average ℓ_1 distance between the populations.

We also present polynomial time algorithms for the resulting max margin which, for now, needs only slightly more data than stated above. Our methods can also be used to give a simple combinatorial algorithm for finding a bisection in a random graph that matches Boppana's convex programming approach (and McSherry's spectral results).

1 Introduction

Recent technology permits large-scale association studies, in which a complex disease such as cancer or Alzheimer's disease is associated with some genetic factors. One of the main difficulties in these studies is to filter out the population effects from the data, so that the studied disease will be the main property affected by the genetic information (this is called population stratification, or population substructure). Mathematically, this boils down to a classification problem, in which a set of bit vectors are classified into clusters. Each vector corresponds to an individual participating in the study, and each bit corresponds to a position in the genome (known as a SNP - single nucleotide polymorphism).

In this paper, we consider a specific scenario of population stratification, in which the population at hand is known to be composed of two sub-populations. We assume a random generative model for the two populations. Each population is represented by a vector, in which every coordinate represents the prior probability of the bit vector to be 1 in that position for this population. We assume that each bit vector is generated according to the population's probability vector. Moreover, we assume the features are chosen independently. [1] We note that similar models have been proposed for many other problems, including, for example, clustering users according to their preferences [13, 14, 6].

The ability to classify the vectors clearly depends on the difference between the probability vectors of the two populations. We therefore measure the difference between populations according to two measures. If $\vec{p_1}$ and $\vec{p_2}$ are the two probability vectors, we denote by γ and α the average ℓ_2^2 and ℓ_1 distances respectively between $\vec{p_1}$ and $\vec{p_2}$. In particular, we define $\gamma = E_f[(p_1(f) - p_2(f))^2]$ and $\alpha = E_f[|p_1(f) - p_2(f)|]$.

Here, we focus on minimizing the amount of data required to cluster these vectors into two clusters. We have two approaches; one is *graph-based* and one is *model-based*. For our graph based approach, we use

[*] Supported by by NSF grant CCF-0105304 and NSF grant EF-0331494. University of Califonia, Berkeley, CA.

[†] Supported by NSF grant IIS-0513599. International Computer Science Institute, Berkeley, CA

[‡] Sponsored in part by the Army Research Office under agreement number DAAD19-02-1-0389 and NSF grant CNF-0435382. Carnegie Mellon University, Pittsburgh, PA

[1] This independence assumption has been referred to as the naive Bayes model.

the fact that humans are *diploid*, that is, in each position, each individual has two bits, representing the two copies of the chromosome (corresponding to the two parents).[2] We define a *dissimilarity* function on pairs of individuals which is based on the diploidy of the human DNA data, and build an associated dissimilarity graph. We assume the scenario where each of the clusters contains N individuals. We show that the optimal solution to a balanced max-cut problem in the dissimilarity graph gives the correct partition with k bits, where $Nk = \Theta(\log N/\gamma^2)$. We note that it is quite straightforward to show that $k = O(\log N/\gamma^2)$ bits are sufficient using a simple connected components algorithm. Our analysis for the max-cut corresponds better to how things work in practice. In particular, some state of the art methods for population stratification (i.e., *STRUCTURE* [19], and spectral based clustering [18]), and our implementations achieve good results when Nk is approximately $1/\gamma^2$ even when k is much smaller than $1/\gamma^2$. We note, however, that this is a structural result; we do not provide a polynomial time algorithm for finding the best cut. Still, we believe it is important to set a rigorous and analytical reasoning for why our method and previous methods [19, 18] work with such little data.

Our second set of algorithms is based on a *model-based* approach where one simultaneously estimates the probabilities of the features and the partition. In this case, we can prove that finding the partition that maximizes the ℓ_1 difference in number of features yields the correct partition with high probability when $Nk = \Theta(\log N/\alpha^4)$. When all the features have the same difference $\gamma = \alpha^2$, and this gives the same bound as the graph-based methods, but for varying features the graph-based method is much better.

On the other hand, we give a model based polynomial time algorithm which finds the correct partition with high probability for some $Nk = \Theta(\log^2 N/\alpha^4)$.[3] That is, we get an algorithm with very nearly the same data requirements as our structural result for this model based approach. We suspect that the graph based approach can be made algorithmic as well.

We note that our algorithms may be of independent interest. As evidence, we give a simple algorithm for graph bisection in the planted bisection model that is competitive with recent results of McSherry [16] and classical results of Boppana [5] for this problem.

1.1 Related Work

Over the last couple of years, a few methods have been suggested for population stratification. The most widely used method, *STRUCTURE* [19], uses a Bayesian model, and an MCMC algorithm to find the correct clustering based on the model. Their method works very well in practice, although it is somewhat inefficient, and there is no rigorous proof that it converges to the correct classification. Recently, a method which is based on spectral analysis was suggested (*EigenStrat* [18]). This method can be viewed as a graph-based method, where the distance measure used is the covariance between the bit vectors. This method performs well on many instances in practice, but it has not been shown rigorously that it will work well under our model.

In general, clustering is a very well studied problem. In particular, there is a large body of work on correlation clustering, [4, 2, 8, 21], where the graph and the optimization problems are given. Unlike these works, we start with a problem and its underlying data and define the graph and the optimization problem based on the characteristics of our data - the challenge is to show that the resulting clustering solves our original problem.

There is ample work on learning mixtures of Gaussians[10, 3, 1, 11, 22] as opposed to the types of distributions we learn. Perhaps the most closely related are the algorithms of Arora and Kannan[3] which can actually learn overlapping Gaussians.

Surprisingly, we observe that the algorithms used in this work are very related to algorithms for collaborative filtering (in which internet users are recommended items for purchase). Previous works on collaborative filtering [13, 14, 15], cluster the items and use the clustering to infer the preferences of the individuals. In contrast, one could view our technique as a way of clustering users according to their preferences.

In fact, Kleinberg and Sandler[13] give an algorithm that uses a quartet based dissimilarity function and computes connected components in a thresholded graph. Our first graph-based approach is highly similar to their method, when the users are viewed as the DNA bits, and the items as the individual vectors.

Another related problem is planted bisection. We note that it is very different in the following respect from our setting. The edges there are essentially generated independently whereas all the edges for a node in our graph are dependent on the feature bits associated with that node, which makes the analysis very different. For the planted bisection problem, there have been a number of results since Boppana's original result [5] which used the ellipsoid method. Many others have since done related work both to understand commonly

[2] Zhou [23] has shown that this requirement is unnecessary using a hamming distance based score along with a max-cut based optimization function.

[3] One can argue that N need only be polynomial in $1/\alpha$ or $1/\gamma$, so the notion of polynomial should be interpreted in terms of these quantities.

used heuristics [12, 7], to produce simpler algorithms [9, 16], or to solve related problems [20, 16].

There is also related work on approximation algorithms for dense instances of graph partitioning as well as more general higher degree polynomial optimization problems. For example, Arora et.al. [3] give algorithms for a broad class of such problems. Here too, the methods appear related to the methods we use here for clustering.

2 Graph Based Clustering: Diploid Setup

In this section, we consider the case where each individual consists of two samples for each feature that are known to be drawn from the same population. As noted in the introduction, this is the case for the population inference problem. The data there has the property that an individual has the feature value of both its parents encoded in its DNA (indeed, it's "own" feature value is unknown.)

In this section, we present a dissimilarity measure for this setup and show a simple connected components algorithm that separates the populations given enough features for each individual.

2.1 The Dissimilarity Score The dissimilarity score between individuals is the difference between the average agreement among the internal feature bits and the average agreement among the feature bits across the pair. For example, if individual 1 has 00 for its diploid value and individual 2 has 11, then the internal bits both agree and the cross bits neither agree. For the case where one individual has feature bit values 01, and the other also has 01 the internal bits do not agree and the cross bits depending on the pairing agree by 2 or by 0, so we give an average score of -1. This is how we define the scores in the table 2.1. For two individuals with feature bit values X and Y, we define $score(X, Y)$, as the sum of the diploid dissimilarities (defined in table 2.1) over the values of the pairs of features bits.

	00	11	01
00	0	2	0
11	2	0	0
01	0	0	-1

Table 1: We only define 01 values since 10 is symmetric.

The following lemma is straightforward.

Lemma 2.1. *The expected score for two individuals from the same population for a feature (which consists of a bit for each parent) is 0, and the expected score for two individuals from γ-different populations is $2\gamma k$ (Technically, there is a distribution of values of γ for different features whose average is γ.)*

The above lemma along with a Chernoff/Hoeffding bound can be used to prove that the following lemma which gives a trivial algorithm for clustering the populations.

Lemma 2.2. *If $k = \Omega(\log N/\gamma^2,)$ all the dissimilarity scores between different population individuals are larger than those between same population individuals by at least $\gamma k/3$ with high probability.*

In this case, a connected components algorithm on a thresholded subgraph will yield the proper partition. Indeed, we remark that this connected components algorithm works even for several populations and for varying numbers of individuals.

In practice, and in our biological application in particular, this could result in a large set of features. We show in the next section that one can reduce the number of features by trading it for extra individuals.

3 Graph Based Partitioning

In this section, we look at input data which has N individuals from population 1 and N individuals from population 2. The connected components algorithm of the previous section did not require this condition. We suspect that other partitioning functions and methodologies could be used to relax this condition.

3.1 Some Notation We use the notation $(\mathcal{P}, \overline{\mathcal{P}})$ to denote the correct partition and $(S, \overline{S})$ to denote an arbitrary cut. In our notation, S always denotes the side of the cut which has at least as many individuals from population 1 as from population 2. We say that a cut $(S, \overline{S})$ with N nodes on both sides has L *swapped nodes* if S has $N - L$ nodes from population 1 and L nodes from population 2. The nodes in S from population 1 and the nodes in $\overline{S}$ from population 2 are called the *unswapped nodes.* All cuts we consider in this section are *balanced*, that is, they have N nodes on either side unless otherwise specified.

We take the score $f(S, \overline{S})$ of a cut to be the sum of the edge scores in the cut, where an edge score is the dissimilarity score defined in the previous section. We show that when the number of features k is sufficient, out of all balanced cuts $(S, \overline{S})$, $(\mathcal{P}, \overline{\mathcal{P}})$ has the maximum score with high probability. We also look at $\mathbf{diff}(S, \overline{S})$ which is difference between the value of f on $(S, \overline{S})$ and the value of f on the perfect partition. In other words, $\mathbf{diff}(S, \overline{S}) = f(S, \overline{S}) - f(\mathcal{P}, \overline{\mathcal{P}})$.

Recall that we use the notation γ to denote $\mathbf{E}_f[(p_1^{(f)} - p_2^{(f)})]^2$. Thus the expected score difference between edges connecting two nodes belonging to different populations is $2\gamma k$, and the expected score on an edge connecting two nodes belonging to the same population is 0.

We sometimes use the notation P_1 to denote the set of individuals in population 1 and P_2 to denote the set of individuals in population 2. We use the term *feature* to denote the data bits available for a single individual and the term *sample* to mean the total feature bits across all individuals.

3.2 Our Results Our main result in this section can be summarized by the following theorem.

Theorem 3.1. *Suppose we have data for the presence or absence of k independent features for individuals belonging to two populations. Also suppose that we have this data for N individuals in each population. If $k \geq \max(\frac{2^{20} \log N \log\log N}{N\gamma^2}, \frac{1024 \log N}{\gamma})$, the correct partition $(\mathcal{P}, \overline{\mathcal{P}})$ has the highest score with probability at least $1 - \frac{1}{N}$.*

A more optimized calculation shows that the constants in the theorem are at most 40 in both expressions. In this paper, we present the proof with higher constants. A complete proof is given in [23]. We provide the main ideas here.

Our proof strategy works roughly as follows. We first show in Lemma 3.1 that with high probability, the average score between any individual and an individual in its own population is lower than the average score between the same individual and an individual in a different population. We then show in Theorem 3.2 that if this event happens for all individuals in the data set, any arbitrary cut $(S, \overline{S})$ has a higher value of f than the perfect partition with at most exponentially low probability. The union bound over all cuts in the graph gives us Theorem 3.1.

We begin with Lemma 3.1. Given an individual u belonging to population 1, let $E(u)$ denote the following event:

$$D = \mathbf{E}_{x \in P_1}[score(u, x)] - \mathbf{E}_{y \in P_2}[score(u, y)] \geq -\gamma k$$

This random variable D has expectation $-2\gamma k$. The event $E(u)$ is the event that it varies (in a bad direction) by at least γk. That is, this choice of individual u is not as close to their own population as is desired. The expectations are taken over random choices of x and y. We define the notion of event $E(u)$ for u in population 2, by reversing the sign in the definition of D above, and proceeding as above.

Let $E = \cup_u E(u)$. We show that if the number of samples is sufficient, E occurs with low probability.

Lemma 3.1. *If $k > \frac{72(c+2)\log N}{\gamma}$, none of the events $E(u)$ occur with probability at least $1 - \frac{1}{N^c}$ over the choice of the feature bits of u for some constant $c > 1$.*

The proof follows by an application of the Method of Bounded Differences and appears in the full manuscript.

For the rest of our proof, we sometimes assume that $\overline{E}$ holds and sometimes that $\overline{E_u}$ holds for $u \in V'$ where V' is some subset of V.

We look at cuts $(S, \overline{S})$ which contain L swapped nodes from the perfect partition. Let us denote the cut by $(X \cup P, Y \cup Q)$. Here X and Y are the sets of unswapped nodes belonging to population 1 and 2 and P and Q are the sets of swapped nodes belonging to population 2 and 1 respectively. On expectation, any such cut has a lower value of the score f than the perfect partition. We would like to show that all such cuts $(S, \overline{S})$ have a lower value of f than the perfect partition with high probability. One way to do this is to show that the variance of the value of f on an arbitrary cut $(S, \overline{S})$ is low. Unfortunately, this is not true. What we can show, however, is that $\mathbf{diff}(S, \overline{S})$, the difference between the scores of cut $(S, \overline{S})$ and the perfect partition has low variance compared to its expectation; the perfect partition therefore has the highest objective value of all cuts with high probability.

Theorem 3.2. *Suppose that the event E does not occur. If $k \geq \max(\frac{2^{20} \log N \log\log N}{N\gamma^2}, \frac{1024 \log N}{\gamma})$ for an arbitrary cut $(S, \overline{S})$ with L swapped nodes,* $\Pr[\mathbf{diff}(S, \overline{S}) \geq 0] \leq \frac{3}{N^{2L+2}}$.

Proof:

The goal of the proof is to look at the function $\mathbf{diff}(S, \overline{S})$ as a martingale. Unfortunately, we cannot get a good bound on the deviation if we do so right away.

The proof therefore proceeds in four main steps. Given the cut $(S, \overline{S}) = (X \cup P, Y \cup Q)$, we first fix a configuration of the feature bits of the individuals in P and Q such that they satisfy certain properties: essentially, that the sum, over featuers, of the deviations from the expected value of the number of individuals in P and Q is small. Lemma 3.3 shows that this fails to happen with only exponentially small probability. Now the function f is a random function of the features of the unswapped nodes. Now we would like to show that if the event E does not occur, the function f is a martingale which has low deviation compared to its expectation. However, doing so is a bit complicated.

To simplify the calculations, we define another event E_S: $E_S = \cup_{u \in P \cup Q} E(u)$.

The event E_S occurs when one of the individuals in the set of swapped nodes is further from the average individual in its own population than from the average individual in a different population. The proof of Lemma 3.1 implies that for all possible subsets of swapped nodes, E_S does not occur with high probability since it is really just a condition on each individual node which holds individually with high probability.

We proceed in the following by conditioning on $\overline{E_S}$. We will complete the proof by showing that the probability is essentially the same in the event that we condition on $\overline{E}$ which is required in Theorem 3.2.

We next show in Corollary 1 that if E_S does not occur, the expectation of $f(S, \overline{S})$ is much less than the expectation of the score f evaluated on the perfect partition. Note that this expectation is taken over the random feature bits of the individuals in X and Y. In Lemma 3.4, we look at the function f as a martingale and show a deviation bound conditioned on $\overline{E_S}$. Finally we demonstrate how this implies a deviation bound on $\mathbf{diff}(S, \overline{S})$ conditioned on $\overline{E}$.

Lemma 3.2. *Suppose event E_S does not occur. Then for any node u in population* 1, $\mathbf{E}_{x \in Q}[score(u,x)|\overline{E_S}] - \mathbf{E}_{y \in P}[score(u,y)|\overline{E_S}] \leq -\frac{\gamma k}{2}$

The lemma says that conditioning on the event $\overline{E_S}$ does not change the expected score difference between an individual u and individuals x and y in different populations.

This follows from the fact that the event E_S occurs with very low probability so that excluding it from the sample space does not change the expectation of this random variable by much. The proof is an the full manuscript.

The following corollary follows from the fact that $\mathbf{diff}(S, \overline{S})$ is the sum over all nodes in X and Y of the quantity in Lemma 3.2, along with linearity of expectation. The proof is in the full manuscript.

Corollary 1. *Suppose that the event E_S does not occur. Then,* $\mathbf{E}[\mathbf{diff}(S, \overline{S})] \leq -\frac{1}{2}\gamma k L(N-L)$.

We also use the notation $q_1^{(f)} = 1 - p_1^{(f)}$ and $q_2^{(f)} = 1 - p_2^{(f)}$. Let the number of nodes in P with a 00 value for feature f be $(q_1^{(f)})^2 L + t_f^{00}\sqrt{L}$. We impose the condition that

$$(3.1) \sum_f (t_f^{00})^2 \quad \leq \quad 4L \log N + 2k \log\log N + O(k)$$

Similarly let t_f^{01} and t_f^{11} be the corresponding deviations in the number of individuals in P with value 01 and 11 for feature f, and s_f^{00}, s_f^{01} and s_f^{11} be the corresponding deviations in the number of individuals in Q with value 00, 01 and 11 for feature f. We also impose the same constraints on these other deviations. Let F denote the event that not all these constraints are satisfied. We show that the probability of F is very low.

Lemma 3.3. $\Pr[F|\overline{E_S}] \leq \frac{1}{N^{2L+2}}$

The proof is in the full manuscript. The idea is that the probability that the number of 00's for a feature f varies from its expectation by $t_f\sqrt{L}$ is at most $e^{-t_f^2}$. Assuming independence of the features (which is not quite true under the condition that E_S does not hold), the probability that overall, the features vary according to $t_1, \ldots, t_k$ is at most $e^{-\sum t_f^2}$. This probability is small enough to tolerate a union bound over configurations for $t_1, \ldots, t_k$, as well as the conditioning on $\overline{E_S}$.

The next lemma shows a bound on the deviation of $\mathbf{diff}(S, \overline{S})$ from its expected value, given that $\overline{E_S}$ and $\overline{F}$ hold.

Lemma 3.4. *Let $(S, \overline{S})$ be an arbitrary cut with L swapped nodes, and let* $t = \max(8L\sqrt{LN\gamma k \log N}, 256L\sqrt{kN \log N \log\log N})$. *Then,* $\Pr[|\mathbf{diff}(S, \overline{S}) - \mathbf{E}[\mathbf{diff}(S, \overline{S})]| > t|\overline{E_S} \cap F] \leq \frac{1}{N^{2L+2}}$.

Proof:

We use the Method of Average Bounded Differences [17], as restated in Theorem 3.3.

Theorem 3.3. *[17] Let $X_1, \ldots, X_n$ be an arbitrary set of random variables and let ϕ be a function satisfying the property that for each $i \in [n]$, there is a non-negative c_i such that*

$$|\mathbf{E}[\phi|X_i] - \mathbf{E}[\phi|X_{i-1}]| \leq c_i$$

Then,

$$\Pr[|\phi - \mathbf{E}[\phi]| > t] \leq 2e^{-t^2/2C}$$

where $C = \sum_{i \leq n} c_i^2$

We apply this lemma to $\mathbf{E}[\mathbf{diff}(S, \overline{S})|\overline{E_S}]$ which we estimate in Corollary 1. This function depends on the choice of the feature bits of the individuals in X and Y. Note that $\mathbf{E}[\mathbf{diff}(S, \overline{S})|\overline{E_S}, \overline{F}]$ does not differ very much from $\mathbf{E}[\mathbf{diff}(S, \overline{S})|\overline{E_S}]$ as F occurs with very low probability.

Now, we let $B_{u,f}$ be the choice of feature f for individual u. We choose these bits in lexicographic order for the sake of using the Method of Bounded differences. We proceed to give an upper bound on $c_{u,f}$ by bounding the *maximum* change in f given the conditions F and $\overline{E_S}$ on the choice of P and Q.

Note that if we conditioned on $\overline{E}$, the feature bits of an unswapped individual u would not vary independently. Conditioning on $\overline{E_S}$ instead ensures that the feature bits of an individual u in $X \cup Y$ are independent and allows us to compute $c_{u,f}$.

When the state of the feature f changes from 11 to 00 in an unswapped individual u, $c_{u,f}$ can be bounded as follows. Note that $\mathbf{diff}(S, \bar{S})$ is precisely the scores of the following edges:

$$\sum_{u \in X} \left(\sum_{q \in Q} score(u,q) - \sum_{p \in P} score(u,p) \right) + \sum_{v \in Y} \left(\sum_{p \in P} score(v,p) - \sum_{q \in Q} score(v,q) \right)$$

From the way we set the feature bits in P and Q, there are $(p_1^{(f)})^2 L + t_f^{11}\sqrt{L}$ nodes in Q with value 11 for feature f, and $(q_1^{(f)})^2 L + t_f^{00}\sqrt{L}$ nodes in Q with value 00 for feature f. Each edge between u and any of the former set of nodes now contribute 2 extra to $\mathbf{diff}(S, \bar{S})$, and each edge between u and each of the nodes in the latter set now no longer contribute 2 to $\mathbf{diff}(S, \bar{S})$. A similar calculation can be done for the edges between u and the nodes in P. In summary, we can bound $c_{u,f}$ as:

$$\begin{aligned} c_{u,f} &\le 2((p_1^{(f)})^2 - (q_1^{(f)})^2 - (p_2^{(f)})^2 + (q_2^{(f)})^2)L \\ &\quad + 2t_f^{00}\sqrt{L} + 2t_f^{11}\sqrt{L} + 2s_f^{00}\sqrt{L} + 2s_f^{00}\sqrt{L} \\ &\le 2(p_1^{(f)} - p_2^{(f)})L + 2\sqrt{L} \\ &\quad (t_f^{00} + t_f^{11} + s_f^{00} + s_f^{11}) \end{aligned}$$

The other cases are similar. For any change in u's feature bits, we can bound $c_{u,f}$ to be at most $2(p_1^{(f)} - p_2^{(f)})L + 2\sqrt{L}(t_f^{00} + t_f^{01} + t_f^{11} + s_f^{00} + s_f^{01} + s_f^{11})$.

The total deviation C can be bounded as

$$\begin{aligned} C &= \sum_f \sum_{u \in X \cup Y} c_{u,f}^2 \\ &\le 8L^2 N\gamma k + \\ &\quad 8LN \sum_f (t_f^{00} + t_f^{01} + t_f^{11} + s_f^{00} + s_f^{01} + s_f^{11})^2 \\ &\le 16L^2 N\gamma k + \\ &\quad 384LN(4L\log N + 2k\log\log N + O(k)) \end{aligned}$$

If the first term dominates the sum, for $t = 8L\sqrt{LN\gamma k \log N}$, $\frac{t^2}{2c} \ge (2L+2)\log N$ and the lemma holds by the method of bounded differences (Theorem 3.3.) If the second term dominates, for $t = 256L\sqrt{kN\log N \log\log N}$, $\frac{t^2}{2c} \ge (2L+2)\log N$ and the lemma holds by Theorem 3.3. □

Lemma 3.4 and Corollary 1 show that conditioned on $\overline{E_S}$ and $\overline{F}$ for any cut $(S, \overline{S})$, $\mathbf{diff}(S, \overline{S})$ is at most $-\frac{1}{2}\gamma kNL + t$ with high probability, where t is defined as in Lemma 3.4. If k is large enough, namely, if $k \ge \max(\frac{2^{20}\log N \log\log N}{N\gamma^2}, \frac{1024\log N}{\gamma})$, t is at most $\frac{1}{2}\gamma kNL$.

Let B be the bad event that $\mathbf{diff}(S, \overline{S}) > 0$. So far we have shown that the probability of B is at most $\frac{1}{N^{2L+2}}$ conditioned on $\overline{E_S}$ and $\overline{F}$. As event $\overline{E}$ implies $\overline{E_S}$, $\Pr[B|\overline{E_S}, \overline{F}] \ge \Pr[B|\overline{E}, \overline{F}]\Pr[\overline{E}|\overline{E_S}, \overline{F}]$. Thus, $\Pr[B|\overline{E}, \overline{F}]$ is less than $\frac{2}{N^{2L+2}}$ since $\Pr[\overline{E}|\overline{E_S}, \overline{F}]$ is close to 1 (bigger than 1/2 is sufficient.) Since $Pr[F] < 1/N^{2L+2}$ is so small, we can conclude that $\Pr[B|\overline{E}]$ remains less than $\frac{3}{N^{2L+2}}$ which proves Theorem 3.2. □

4 Model-Based Partitioning

In this section, we consider partitioning using a model-based approach. A model-based approach is one in which we use the data to simultaneously estimate the probability of occurrence of each feature in the two populations as well as assign individuals to populations.

We consider two model-based approaches. First, we present a simple combinatorial algorithm which runs in polynomial time and finds the perfect partition when k and N are large enough. Next, we define a model-based score g on the partitions of the individuals. We show that the partition which maximizes the score g is the correct partition with high probability, provided k and N are large enough. However we do not know how to compute this partition efficiently.

4.1 Preliminaries We use the following notation, some of which we defined in Section 3. We use $p_1^{(f)}$ and $p_2^{(f)}$ to denote the probabilities of occurrence of feature f in populations 1 and 2 respectively. For a specific feature f, we let $\alpha_f = |p_1^{(f)} - p_2^{(f)}|$ and we let $\alpha = \mathbf{E}_f[|p_1^{(f)} - p_2^{(f)}|]$, so that $\sum_f \alpha_f = \alpha k$.

For the rest of the section, we only look at partitions which have N nodes on either side. The algorithm of this section, iteratively produces better and better partitions. We always assume without loss of generality that S has at least as many individuals from population 1 as does $\bar{S}$. A partition $(S, \bar{S})$ has *advantage* ϵ if S has $(\frac{1}{2} + \epsilon)N$ individuals from population 1.

Given a partition $(S, \bar{S})$ with positive advantage, let $\tilde{p}_S^{(f)}$ and $\tilde{p}_{\bar{S}}^{(f)}$ denote the fraction of individuals with feature f in S and $\bar{S}$ respectively. We say that feature f is *in the right order* if $\tilde{p}_S^{(f)}$ and $\tilde{p}_{\bar{S}}^{(f)}$ are in the same order as $p_1^{(f)}$ and $p_2^{(f)}$.

We now show an useful lemma, which, given a partition $(S, \bar{S})$ with advantage ϵ, determines the probability with which a feature f is in the right order in this partition.

Lemma 4.1. (Ordering Lemma) *Let f be a feature such that $p_1^{(f)} < p_2^{(f)}$ and $(S, \bar{S})$ be a partition of $2N$ individuals with advantage ϵ. If $2\epsilon\alpha_f < \frac{1}{\sqrt{N}}$,*

$$\Pr[\tilde{p}_S^{(f)} < \tilde{p}_{\bar{S}}^{(f)}] \geq \frac{1}{2} + 2\epsilon c^2 \alpha_f \sqrt{N}$$

Otherwise,

$$\Pr[\tilde{p}_S^{(f)} < \tilde{p}_{\bar{S}}^{(f)}] \geq 1 - e^{-\epsilon^2 \alpha_f^2 N}$$

where c is a constant.

Proof: Let $N_f(S)$ and $N_f(\bar{S})$ be the number of people with feature f in S and $\bar{S}$ respectively. Then, $N_f(S)$ and $N_f(\bar{S})$ are sums of N independent 0/1 random variables, with $\mathbf{E}[N_f(S)] = [(\frac{1}{2}+\epsilon)p_1^{(f)} + (\frac{1}{2}-\epsilon)p_2^{(f)}]N$ and $\mathbf{E}[N_f(\bar{S})] = [(\frac{1}{2}+\epsilon)p_2^{(f)} + (\frac{1}{2}-\epsilon)p_1^{(f)}]N$.

The difference between the expectations of the two distributions is $2\epsilon\alpha_f N$. When $2\epsilon\alpha_f \leq \frac{1}{\sqrt{N}}$, we can apply Lemma 4.2 to get the first bound. When $2\epsilon\alpha_f > \frac{1}{\sqrt{N}}$, the lemma follows by an application of the Hoeffding Bound. □

Lemma 4.2. *Let W_1 and W_2 be sums of n 0/1 independent random variables with $0 < \mathbf{E}[W_2] - \mathbf{E}[W_1] < \sqrt{n}$. Then,*

$$\Pr[W_2 \geq W_1] \geq \frac{1}{2} + \frac{c^2 G}{\sqrt{n}}$$

where $G = \mathbf{E}[W_2] - \mathbf{E}[W_1]$ and c is a constant.

We defer the proof of this lemma to the full version of the paper. We use the following version of the Hoeffding Bound in a later part of the section.

Theorem 2. (Hoeffding Bound) *[17] Let $X_1, \ldots, X_n$ be independent random variables such that X_i lies between a_i and b_i, and let $X = \sum_{i=1}^n X_i$. Then,*

$$\Pr[|X - \mathbf{E}[X]| > t] \leq 2e^{-t^2/2\sigma}$$

where $\sigma = \sum_i (a_i - b_i)^2$.

4.2 A Model-Based Algorithm In this section, we describe our model-based algorithm.

The intuition of our algorithm is as follows. Suppose we were given the correct classification of a constant fraction of the nodes. Then we could estimate with reasonable accuracy the probability of the presence of each feature in either population and use a simple scoring function to find the correct side for each unclassified individual. In reality, we start with a random (though slightly biased) partition and repeatedly (re)classify to get more and more correct partitions.

In the beginning, we divide the set of all individuals randomly into two equally large sets, which we call the training set T and the test set T'. Let $2n = N$ be the size of each such set. We begin with a random partition of the individuals in the training set into two equal sized sets S_0 and $\bar{S}_0$. Notice that with at least constant probability, the two sides are biased toward one population or another by an additive $\sqrt{n}$ individuals. We estimate the probabilities for a fixed subset of the features in each of S_0 and $\bar{S}_0$. Using these probabilities, we estimate using a simple scoring function which side each individual in the test set is more likely to have come from.[4] We use this score to form a new partition $(S_1, \bar{S}_1)$ of the test set and repeat the learning and partitioning process on alternative halves for $\Theta(\log n)$ iterations as shown below. At this point, the classification would be approximately correct and it is easy to now find a correct classification.

In each subsequent learning/categorization phase, we use a separate set of features to maintain independence between the phases. We randomly divide the set of all features into $\Theta(\log n)$ groups, and use a new group for each round. Standard techniques can be used to show that the value of α and γ when restricted to each of these groups is within a constant factor of α and γ; for ease of exposition we simply assume from now on that they are equal. We also abuse notation and use k to denote the number of features in each round; the total number of features we need is thus $\Theta(k \log n)$.

We compute our scoring function as follows. We initialise the score of an individual to 0 and look at each feature in order. For each feature f,

- if $\tilde{p}_1^{(f)} > \tilde{p}_2^{(f)}$, add 1 to the score if the feature is present, and otherwise subtract 1.
- if $\tilde{p}_1^{(f)} < \tilde{p}_2^{(f)}$, subtract 1 from the score if the feature is present, and otherwise add 1

We denote the result by $score(x)$ for an individual x. The following theorem summarises the main result of this section.

Theorem 4.1. *Suppose we have data on the presence or absence of $\tilde{k}$ features in N individuals. Then, with probability at least $1 - \frac{1}{N}$, our algorithm finds the correct partition in $\Theta(\log N)$ rounds if $\tilde{k} > \Theta(\frac{\log^2 N}{\alpha^2})$ and $N\tilde{k} > \Theta(\frac{\log^2 N}{\alpha^4})$.*

If x and y are individuals from populations 1 and 2 respectively, the goal of the algorithm is to tell x and y apart based on their scores. For this, we would like to estimate how the quantity $score(x) - score(y)$ behaves. Note that this quantity is not only a function of the randomness in the features of x and y but it also depends on the randomness in the feature bits of the individuals in the training set. The next two

[4] We do not, use the technical notion of likelihood, though.

lemmas ensure that with high probability over the randomness in the training set, the expected value of $score(x) - score(y)$ is high, where the expectation is taken only over the feature bits of x and y.

For a given advantage ϵ, we call a feature f a *low feature* if $2\epsilon\alpha_f < \frac{1}{\sqrt{n}}$, and a *high feature* otherwise. For the rest of the section, we use $\mathcal{L}$ and $\mathcal{H}$ to denote the set of low and high features respectively. We also say that the low features dominate when $\sum_{f\in\mathcal{L}} \alpha_f \geq \frac{\alpha k}{2}$; otherwise, we say that the high features dominate.

Lemma 4.3. *Let Z_f be a random variable such that $Z_f = \alpha_f$ if the feature f is in the right order and $-\alpha_f$ otherwise, and let $Z = \sum_f Z_f$. Also let $\epsilon\sqrt{n} > c'$, where c' is a constant and $k > \Theta(\frac{\log n}{\alpha^2})$. When the high features dominate,*

$$\Pr[Z < \frac{\beta\alpha k}{4}] \leq \frac{1}{n}$$

Otherwise,

$$\Pr[Z < \frac{\epsilon c^2\alpha^2 k\sqrt{n}}{4}] \leq \frac{1}{n}$$

Here c is the constant in Lemma 4.2 and β is a constant.

The proof uses Lemma 4.1 along with Theorem 2. We omit the details here.

Lemma 4.4. *Let x be a node belonging to population 1 and y be a node belonging to population 2. Let $(S_i, \bar{S}_i)$ be a partition with advantage ϵ. If the low features dominate, with probability at least $1 - \frac{1}{n}$ over the feature bits of the individuals in the training set,*

$$\mathbf{E}[score(x)] - \mathbf{E}[score(y)] \geq \frac{\epsilon c^2\alpha^2 k\sqrt{n}}{4}$$

Otherwise,

$$\mathbf{E}[score(x)] - \mathbf{E}[score(y)] \geq \frac{\beta\alpha k}{4}$$

where c' is a constant, c is the constant in Lemma 4.2 and β is the constant in Lemma 4.3. The expectation here is taken over the choice of the feature bits for nodes x and y.

The basic idea is to observe that the difference between the scores is basically twice the score Z in Lemma 4.3. We omit the details.

Lemma 4.5. *Let $(S_i, \bar{S}_i)$ be a partition with advantage at least ϵ where $\epsilon\sqrt{n} > c'$, and let $k > \Theta(\frac{\log n}{\alpha^2})$ and $nk > \Theta(\frac{\log n}{\alpha^4})$. Then, with probability at least $1 - \frac{1}{n}$, $(S_{i+1}, \bar{S}_{i+1})$ has advantage at least $\min[2\epsilon, C_0]$, where C_0 is a constant.*

We leave the proof of this lemma to the full manuscript. For now, we observe Lemma 4.3 holds with high probability for every step of the algorithm. Then, for each node in the test set, we have a lower bound on its expected score. Moreover, we know that the variance is at most $\sqrt{k}$. Using Lemma 4.2, we can infer a lower bound on the probability of categorizing a test node correctly. If we are in the first case of Lemma 4.3, this lower bound becomes $\frac{1}{2} + 2\epsilon$. This yields a new partition with the required 2ϵ advantage. In the other case of Lemma 4.3, we can get a constant advantage.

The correctness of our algorithm is ensured by the following final lemma, the proof of which we defer to the full version of the paper.

Lemma 4.6. *Let $(S_i, \bar{S}_i)$ be a partition of the training set which has advantage at least C_0 where C_0 is a constant. If $k > \Theta(\frac{\log n}{\alpha^2})$, we can correctly classify every node in the test set with probability at least $1 - \frac{1}{n}$.*

4.3 A Model-Based Optimization Function We now introduce our model-based optimization function. Let $N_f(S)$ denote the number of individuals with feature f in the set of individuals S. Given a partition $(S, \bar{S})$ of the individuals, our score $g(S, \bar{S})$ is defined as follows:

$$g(S, \bar{S}) = \sum_f \left|N_f(S) - N_f(\bar{S})\right|$$

We also define $\mathbf{diff}_g(S, \bar{S})$, which is the difference between $g(\mathcal{P}, \bar{\mathcal{P}})$ and $g(S, \bar{S})$. The main result of the section can be summarised by the following theorem.

Theorem 4.2. *Suppose we are given data for the presence or absence of k features in N individuals from two populations. If $k > \Theta(\frac{\log N}{\alpha^2})$ and $N > \Theta(\frac{\log N}{\alpha^2})$, the partition $(S, \bar{S})$ with N nodes on each side that maximizes the value of the score $g(S, \bar{S}) = \sum_f \left|N_f(S) - N_f(\bar{S})\right|$ is the correct partition with probability at least $1 - \frac{1}{N}$.*

Proving Theorem 4.2 is easy when L is comparable to N, say $L > N/4$. When $N > \Theta(\frac{\log N}{\alpha^2})$, we can show that the expected difference between the values of the perfect partition and an arbitrary partition $(S, \bar{S})$ with L swapped nodes is at least $\alpha k L/5$. We omit the details here. Now the difference in the scores is a function of $Nk = \Theta(Lk)$ bits, and if we change the value of each of these bits, the score changes by at most 2. Therefore the probability that this difference is less than 0 is at most the probability that it differs from its expected value by $\Theta(L\sqrt{k\log N})$, and by the Method of Bounded Differences, this value is exponentially small in N.

We cannot use this argument directly when $L \leq N/4$ because now the expected value of the difference is $\Theta(\alpha k L)$ which can be much smaller than the standard deviation $\sqrt{Nk}$. What we can do however, is show that the difference behaves well for most configurations of the feature bits of the unswapped individuals.

For the rest of the section, we focus on proving theorem 4.2 when $L < N/4$. Using notation from Section 3, let $(S, \bar{S})$ be the partition $(X \cup P, Y \cup Q)$, where all nodes in X and Q are in population 1 and all nodes in Y and P are in population 2. The main idea behind the proof is to look at only a region of the total probability space on which a certain condition holds. Let $\tilde{p}_X^{(f)}$ and $\tilde{p}_Y^{(f)}$ be the fraction of individuals who have feature f in the sets X and Y respectively. Let S_f be a random variable which is 1 when $\tilde{p}_X^{(f)}$ and $\tilde{p}_Y^{(f)}$ are in the right order and when $|\tilde{p}_X^{(f)} - \tilde{p}_Y^{(f)}| > \frac{\alpha_f}{2}$, and -1 otherwise. The condition we need is the following:

$$\sum_f S_f \alpha_f \geq \frac{8\alpha k}{10} \qquad (4.2)$$

The following lemma shows that this condition holds with probability at least $1 - \max[\frac{1}{N^{2L}}, \frac{1}{2^N}]$ when N and k are large enough.

Lemma 4.7. *Let $N > \Theta(\frac{\log N}{\alpha^2})$ and $k > \Theta(\frac{\log N}{\alpha^2})$ and $L < \frac{N}{4}$. Then, condition 4.2 holds with probability at least $1 - \max[\frac{1}{N^{2L}}, \frac{1}{2^N}]$ over the feature bits of the individuals in X and Y.*

The basic idea of the proof is to use Lemma 3.3 which bounds the sum of squares of the deviation of the number of individuals with feature f from their expected value. An application of the conditions on N and k then completes the proof. The following lemma shows that $\mathbf{E}[\mathbf{diff}_g]$ is high when Condition (1) is satisfied.

Lemma 4.8. *Let $L < N/4$ and let C be a configuration of the feature bits of the individuals in X and Y such that condition 4.2 holds. Then, $\mathbf{E}[\mathbf{diff}_g(S, \bar{S})|C] \geq \frac{\alpha k L}{8}$.*

Proof: Let Δ_f denote the contribution of feature f to $\mathbf{diff}_g(S, \bar{S})$. We make the following two claims, the proofs of which we defer to the full version of the paper.

Claim 3. *Let f be a feature such that $\alpha_f \geq \frac{\alpha}{20}$ and suppose that X_f and Y_f are fixed so that $S_f = 1$. Then, $\mathbf{E}[\Delta_f] \geq \frac{\alpha_f L}{2}$, where the expectation is taken over the randomness in the feature bits of P and Q.*

Claim 4. *Let X_f and Y_f be fixed to any values. Then,$\mathbf{E}[\Delta_f] \geq -\alpha_f L$, where the expectation is taken over the randomness in the feature bits of the individuals in P and Q.*

Let $\mathcal{B}$ be the set of features such that $S_f = 1$ and $\alpha_f \geq \frac{\alpha}{20}$. Then, the expectation of $\mathbf{diff}_g(S, \bar{S})$ can be written as follows.

$$\mathbf{E}[\mathbf{diff}_g(S, \bar{S})] = \mathbf{E}[\sum_f \Delta_f] = \sum_{f \in \mathcal{B}} \mathbf{E}[\Delta_f] + \sum_{f \notin \mathcal{B}} \mathbf{E}[\Delta_f]$$

A feature f is in $\mathcal{B}$ if both $\alpha_f > \frac{\alpha}{20}$ and $S_f = 1$. Condition (1) tells us that $\sum_{\{f|S_f=1\}} \alpha_f$ is at least $\frac{8\alpha k}{10}$ and the total contribution from the features which have $\alpha_f < \frac{\alpha}{20}$ can be at most $\frac{\alpha k}{20}$. Therefore $\sum_{f \in \mathcal{B}} \alpha_f \geq \sum_{\{f|S_f=1\}} \alpha_f - \frac{\alpha k}{20} \geq \frac{15\alpha k}{20}$. Again, a feature is not in $\mathcal{B}$ if either $S_f = -1$ or $\alpha_f < \frac{\alpha}{20}$. $\sum_{\{f|S_f=-1\}} \alpha_f \leq \frac{2\alpha k}{10}$ by Condition (1), and therefore $\sum_{f \notin \mathcal{B}} \alpha_f \leq \frac{2\alpha k}{10} + \frac{\alpha k}{20} \leq \frac{5\alpha k}{20}$.

Applying Claims 4 and 3, for a fixed C such that Condition (1) holds, we can estimate $\mathbf{E}[\mathbf{diff}_g(S, \bar{S})|C]$ to be at least $\frac{15\alpha k}{20} \cdot \frac{L}{2} - \frac{5\alpha k}{20} \cdot L = \frac{\alpha k L}{8}$.

□

The next lemma shows that $\mathbf{diff}_g(S, \bar{S}) < 0$ with probability at most $\max[\frac{2}{N^{2L}}, \frac{2}{2^N}]$ when $(S, \bar{S})$ is a cut with L swapped nodes.

Lemma 4.9. *Let $L \leq N/4$ and let $(S, \bar{S})$ be a cut with L swapped nodes. Then*

$$\Pr[\mathbf{diff}_g(S, \bar{S}) \leq 0] \leq \max[\frac{2}{2^N}, \frac{2}{N^{2L}}]$$

The basic idea behind the proof is as follows. If the feature bits of the individuals in X and Y are fixed, $\mathbf{diff}_g(S, \bar{S})$ is a random function of the features of the individuals in P and Q, and there are only $2Lk$ such bits. When the feature bits of the individuals in X and Y are fixed such that condition (1) holds, Lemma 4.8 shows that the expectation of $\mathbf{diff}_g(S, \bar{S})$ is also high. Finally, since condition (1) holds with high probability, we have Lemma 4.9.

Theorem 4.2 now follows by taking an union bound over all cuts. We omit the details here.

5 Bisection

The techniques of the previous section can be used to give a simple algorithm for bisection in the randomized model introduced by Bui. This matches the result of Boppana which used convex programming. We note that McSherry also matched this result using spectral methods. Our algorithm is a local improvement method of a sort. It is related to the approach of Condon and Karp, or an improved version due to Carson and Impagliazzo.

Suppose we are given a random graph generated as follows. The graph has $2n$ nodes and m edges, and the nodes can be partitioned into two sets of equal size. Each node chooses $\frac{m}{2n}$ neighbors at random as follows: it selects each node on the same side of the partition with probability $\frac{1}{2} + \delta$ and each node on the opposite side with probability $\frac{1}{2} - \delta$ where $\delta = \beta\sqrt{\frac{n}{2m}\log n}$ and β is a constant.

Given a partition $(S, \bar{S})$, we always assume that S has at least as many nodes from the first side as from the second side of the partition. We say that $(S, \bar{S})$ has imbalance ϵ if S contains $(\frac{1}{2} + \epsilon)n$ nodes from the first side. Our algorithm starts with a random partition of the nodes. Note that this partition has imbalance at least $\frac{1}{\sqrt{n}}$ with constant probability. We iteratively reclassify the nodes and produce partitions with increasing imbalance up to a constant imbalance.

For each node we examine the next $d = \frac{m}{2n \log n}$ unexamined (in any previous iteration of the algorithm) edges. We place the node on the side with the majority of neighbors, breaking ties at random. The following lemma shows that when $\epsilon < c'$, where c' is a constant, the imbalance increases by at least a constant factor every round.

LEMMA 5.1. *Let $(S, \bar{S})$ be a partition of the nodes with imbalance ϵ. If $\epsilon < c'$, a node is classified correctly with probability at least $\frac{1}{2} + 4\epsilon$. Here c' is a constant.*

The proof uses ideas similar to the proof of Lemma 4.1.

Since we start with imbalance $\frac{1}{\sqrt{n}}$, we get a partition with constant imbalance after at most $\frac{1}{2} \log n$ rounds. We now use Lemma 5.2 and the remaining $\frac{m}{4n}$ unexamined edges per node to classify each nodes correctly with high probability.

LEMMA 5.2. *Suppose we have a partition $(S, \bar{S})$ with constant imbalance, and $\frac{m}{4n}$ unexamined edges for each node. Then, we can classify each node correctly with probability at least $1 - \frac{1}{n}$.*

The proof is very similar to the proof of Lemma 4.6 and we defer it to the full version of the paper.

Acknowledgements S. Zhou thanks Avrim Blum for many delightful discussions on this topic. We thank Kunal Talwar for suggesting using the Method of Bounded Differences and pointing us to [17].

References

[1] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *Proceedings of the 18th Annual Conference on Learning Theory*, pages 458–469, 2005.

[2] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering. In *Proceedings of 37th ACM Symposium on Theory of Computing*, pages 684 – 693, 2005.

[3] S. Arora and R. Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of 33rd ACM Symposium on Theory of Computing*, pages 247–257, 2001.

[4] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. In *Proceedings of the 43th IEEE Symposium on Foundations of Computer Science*, pages 238–247, 2002.

[5] R. Boppana. Eigenvalues and graph bisection: An average case analysis. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pages 280–285, 1987.

[6] John Canny. Collaborative filtering with privacy via factor analysis. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 238–245, 2002.

[7] Ted Carson and Russell Impagliazzo. Hill-climbing finds random planted bisections. In *Proceedings of SODA 2001*, 2001.

[8] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *FOCS*, pages 524–533, 2003.

[9] Anne Condon and Richard M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

[10] A. Dasgupta, J. Hopcroft, J. Kleinberg, and M. Sandler. On learning mixtures of heavy-tailed distributions. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science*, pages 491–500, 2005.

[11] S. Dasgupta. Learning mixtures of gaussians. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer S cience*, pages 634–644, 1999.

[12] Mark Jerrum and Gregory Sorkin. Simulated annealing for graph bisection. In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pages 94–103, 1993.

[13] J. Kleinberg and M. Sandler. Convergent algorithms for collaborative filtering. In *Proc. 4th ACM Conference on Electronic Commerce*, 2003.

[14] J. Kleinberg and M. Sandler. Using mixture models for collaborative filtering. In *Proceedings of the 36th ACM Symposium on Theory of computing*, pages 569–578, 2004.

[15] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Recommendation systems: A probabilistic analysis. In *Proceedings of the 39th IEEE Symposium on Foundations of Computer Science*, pages 664–673, 1998.

[16] Frank McSherry. Spectral partitioning of random graphs. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer S cience*, pages 529–537, 2001.

[17] A. Panconesi and D. Dubhashi. Concentration of measure for the analysis of randomised algorithms. Draft.

[18] Alkes L. Price, Nick J. Patterson, Robert M. Plenge, Michael E. Weinblatt, Nancy A. Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38(8):904–909, July 2006.

[19] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:954–959, June 2000.

[20] Dekel Tsur Ron Shamir, Roded Sharan. Cluster graph modification problems. In *Proceedings of the 28th International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 379–390, 2002.

[21] C. Swamy. Correlation clustering: maximizing agreements via semidefinite programming. In *Proceedings of SODA*, 2004.

[22] V. Vempala and G. Wang. A spectral algorithm of learning mixtures of distributions. In *Proceedings of the 43rd IEEE Symposium on Foundations of Computer Science*, pages 113–123, 2002.

[23] Shuheng Zhou. *Routing, Disjoint Paths and Classification.* PhD thesis, Carnegie Mellon University, 2006. CMU-PDL-06-109.