Clustering via Matrix Exponentiation

by

Hanson M. Zhou

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

[February 2004]
September 2003

Author .... [illegible]
Department of Electrical Engineering and Computer Science
Sept. 26th, 2003

Certified by .... [illegible]
Santosh Vempala
Associate Professor
Thesis Supervisor

Accepted by .... [illegible]
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Clustering via Matrix Exponentiation

by

Hanson M. Zhou

Submitted to the Department of Electrical Engineering and Computer Science
on Sept. 26th, 2003, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science

Abstract

Given a set of n points with a matrix of pairwise similarity measures, one would like to partition the points into clusters so that similar points are together and different ones apart. We present an algorithm requiring only matrix exponentiation that performs well in practice and bears an elegant interpretation in terms of random walks on a graph. Under a certain mixture model involving planting a partition via randomized rounding of tailored matrix entries, the algorithm can be proven effective for only a single squaring. It is shown that the clustering performance of the algorithm degrades with larger values of the exponent, thus revealing that a single squaring is optimal.

Thesis Supervisor: Santosh Vempala
Title: Associate Professor

Acknowledgments

This is joint work with David Woodruff, who contributed much of the work on the order of magnitude calculation for the error, hours of thoughtful collaboration, and plenty of good laughs. I would also like to thank Jonathan Kelner for some interesting discussions and helpful insights. Finally, I would like to thank my advisor Santosh Vempala for introducing me to this problem and for his wise and merciless guidance.

Contents

Chapter 1

A Clustering Algorithm

1.1 Introduction

Similarity-based clustering partitions a set of points given a matrix of pairwise similarities and finds application in many important problems. One motivating example is clustering web search results. A search for "jaguar" may return numerous pages relevant to either the car or the cat. Given counts of links between pairs of pages as an indicator of similarity, one would like to group the car results together and the cat results together. In the most general form, we are given a set of n points and a matrix M, where M_{ij} gives the distance or similarity between points i and j. The goal is to partition the points such that similar points are grouped together and different points apart.

For example, Kannan, Vetta, and Vempala define a bi-criterial measure of cluster quality in which the number of clusters is to be minimized while maximizing the minimum cluster conductance [9]. This reflects a desire to keep the number of groups small, while maintaining a high degree of similarity within each group.

Previously, Papadimitriou, et. al., proved theoretical guarantees for classifying documents to the correct topic under certain assumptions about topic purity and term overlap, via spectral methods[13]. Azar, et. al., undertake a similar task, but

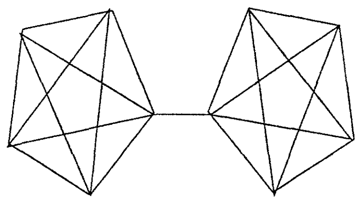

Figure 1-1: Dumbbell

introduce a more general data mining model[1]. In a different vein, Drineas, et. al., give an approximation algorithm for clustering points in Euclidean space so as to minimize the sum of distances squared to each cluster center by first solving a continuous relaxation of the problem using the SVD [5].

1.2 Example

The above graph shows a "dumbbell", in which there are 2 cliques connected by an edge. Partitioning into clusters of high connectivity would yield each of the 2 cliques as a cluster. More generally, each of the parts could be somewhat less densely connected, and the bridging edges could be somewhat more numerous, but still sparse relative to the connectivity within each part.

We would like an algorithm that could classify the vertices into the desired clusters with good probability, for suitable ranges of intra- and inter- cluster edges.

1.3 The Algorithm

We propose the following algorithm for clustering a set of n points with pairwise similarities. For a symmetric matrix M, let M_k^t denote the k-th row(or column) of M^t.

```
Algorithm
Input: A symmetric $\hat{A}$ encoding the pairwise distances between nodes
Output: A partitioning of the nodes into clusters

1. Select some appropriate exponent t
2. Select some appropriate threshold $\epsilon$
3. Compute $\hat{A}^t$
4. For each pair of nodes i,j
    • if $||\hat{A}^t_i - \hat{A}^t_j||^2 < \epsilon$, then i and j are in the same cluster
    • else, i and j are in different clusters
```

Naturally, this leads to the question of how to select the values in steps 1 and 2. In fact, the very effectiveness and performance of the algorithm hinges on using the right t, and depending on t, the right ϵ. We give theoretical results to show that we can correctly cluster for t=2 so long as a certain probability gap under a certain generative model for $\hat{A}$ is $\Omega(\frac{1}{\sqrt{n}})$. In fact, t=2 turns out to be the optimal value of t in some sense, and we demonstrate that the required gap only becomes larger for greater t. We support our theoretical results with experimental evidence. It is important to note that this algorithm is independent of any generative model and is applicable to any arbitrary matrix of similarities $\hat{A}$, for any measure of similarity, generated from any(possibly randomized) process.

1.4 The Random Walk Connection

We have exhibited an algorithm that aims to cluster via only matrix exponentiation. For only t=2, we will demonstrate correct learning of a hidden partition in expectation for a probability gap of $\Omega(\frac{1}{n^{1/4}})$. Experimental evidence indicates that the clustering capability improves initially for larger values of t, but then degrades beyond a certain point. This is expected as one might see by considering a transition matrix M derived from a graph.

It is well known that for any unit vector x corresponding to start position, $x^T M^t$ gives the probabilities of being at a node i after t steps of a random walk on the graph, where each step selects an out-going edge with probability proportional to the edge weight. Moreover, as $t \to \infty$, $\pi = x^T M^t$ gives the stationary distribution, and $\pi_i = \frac{deg(i)}{2m}$, where deg(i) is the degree(sum of incident edge weights) of i, and m is the total weight of all edges [11]. Thus, for t large enough, our algorithm merely elucidates the differences in degrees, which sheds little light on the actual clusters, unless nodes from different parts have significantly different degrees.

As may be suggested by the above, there is a pleasing interpretation of the algorithm in the context of random walks. Viewing $\hat{A}$ as a transition matrix and letting e_i be the unit vector with 1 in the i-th position, $\hat{A}_i^t = e_i^T \hat{A}^t$ gives the probability distribution on the position of a random walk starting from node i after t steps. From this we see that the algorithm makes a pairwise grouping decision based on the L2 distance between the probability distributions after t steps of random walks starting from i and j in order to classify nodes i and j as being similar or different. As discussed above, for very large values of t, all probability distributions will be very close to each other, but as proven for t=2 and experimentally suggested for other reasonably small values of t, the probability distributions of pairs of nodes from the same cluster will converge quicker than those of pairs from different clusters. This is the phenomenae that the algorithm exploits to recover the partitioning.

Chapter 2

Performance Guarantees

We approach this problem from the viewpoint of learning planted partitions from a mixture model. A *mixture model* is a partitioning of a set of nodes into clusters. Pairs of nodes from the same cluster have an edge between them with probability q, and pairs from different clusters have an edge with probability p, where $q > p$.

Let $\hat{A}$ be an n by n matrix of Bernoulli random variables, and $A = E[\hat{A}]$ be the matrix of expectations. The mixture model may be represented by an n by n matrix of random variables $\hat{A}$ with expectation matrix A, where $A_{ij} = q$ for (i,j) in the same cluster, and $A_{ij} = p$ for (i,j) in different clusters. A has k distinct rows corresponding to the existence of k different clusters. The randomized rounding of $\hat{A}$ preserves symmetry: $\hat{A}_{ij} = \hat{A}_{ji}$. For convenience, we denote the cluster of a node i by $\Psi(i)$. WLOG, A is a block diagonal matrix, and $\Psi(i)$ will refer to the cluster of i or the matrix block of i depending on context.

We receive as input an instantiation of the matrix of random variables $\hat{A}$ specified by the mixture model. Henceforth, we will refer to both the matrix of random variables and its instantiation as $\hat{A}$, and it should be clear from the context which is intended. Given this input matrix, our goal is to partition the rows so that a pair of rows are placed in the same partition if and only if they belong to the same cluster. In other words, given $\hat{A}$, recover Ψ.

The clustering is easy to see in the expected matrix A. However, we are not given A, but rather a perturbed version of A through randomized rounding. Fortunately, this graph is not entirely random, as the desired partitions have been "planted" in some sense, by setting the probabilities appropriately according to the mixture model. Intuitively, we see that larger values of q-p make the partitions easier to learn, as larger gaps cause similar points to be better connected relative to dissimilar points. Similarly, larger values of n make learning easier as we have more samples to learn from.

Previous work have made use of this mixture model or special cases of it. Boppana gave a spectral algorithm for the problem of graph bisection on randomly generated graphs, though he requires the solution to a convex optimization problem [3]. Blum and Spencer k-color a randomly generated k-colorable graph so long as $p \geq n^{\epsilon-1}$. They also consider a semi-random model in which a graph generated by an adversary is subject to a small probability of toggling an edge [2]. Condon and Karp partition a random graph into k equal parts, minimizing the number of edges across parts with high probability, so long as $q - p \geq n^{-\frac{1}{2}+\epsilon}$ [4]. Jerrum and Sorkin resolve an open problem of Boppana and Bui by optimally bisecting a random graph with high probability so long as $q - p = \Omega(n^{\delta-2})$, $\delta \leq 2$, via simulated annealing [8].

Finally, McSherry presents an algorithm to learn a hidden partition in a random graph with high probability so long as $q - p = \Omega(\frac{1}{\sqrt{n}})$. The procedure involves a randomized splitting of the columns into two parts and projecting on to the top singular vectors of each part to preserve certain independence properties.

Here, we give a simpler algorithm involving only matrix exponentiation that performs reasonably well even for a single squaring of $\hat{A}$. We show that in fact, t=2 is the optimal exponent, and that larger values of t only asymptotically degrade the performance.

Ultimately, we will provide the following guarantee:

Theorem. *For t=2, the algorithm correctly clusters $1-\delta$ of the rows with probability at least $\frac{1}{2}$, so long as $|q-p| > \frac{2\sqrt{q}(\frac{k^3}{\delta})^{\frac{1}{4}}}{n^{\frac{1}{4}}}$.*

We consider the special case of k equal sized blocks of size s each. We will show in a later section that the case of unequal blocks does not deviate too far from the case of equal blocks, and the asymptotics of the performance guarantees given remain the same so long as the minimum block size s_{min} is a constant fraction of n.

2.1 Preliminaries

Lemma 1. *Let A be an n by n block diagonal matrix of k equal sized blocks of size s each, where $A_{ij} = q$ when $\Psi(i) = \Psi(j)$, and $A_{ij} = p$ when $\Psi(i) \neq \Psi(j)$. If $q > p > 0$, then A has k non-zero eigenvalues: a largest eigenvalue of $sq + (n-s)p$, and an eigenvalue of $(q-p)s$ with eigenspace of dimension k-1.*

Proof. First we will prove that A has rank k. Clearly it has at most rank k as there are only k distinct rows. Suppose A has lesser rank. Let w_i be the length n vector in which every entry is p except for the i-th block of s entries, which are q. We can find constants c_i, not all zero, such that $\sum_{i=1}^{k} c_i w_i = 0$. This implies a series of k equations $c_i q + p\sum_{j=1}^{k} c_j - c_i p = 0$. Summing these k equalities, we get $q\sum_i c_i + (k-1)p\sum_i c_i = 0$. If $\sum_i c_i \neq 0$, then this gives $q = -(k-1)p$. Otherwise, WLOG, $c_1 \neq 0$ and we get $c_1 q - c_1 p = 0 \Rightarrow q = p$. Both cases violate $q > p > 0$ and so A has rank k.

Let λ be an eigenvalue of A, and v the corresponding eigenvector. $Av = \lambda v \Rightarrow \lambda v_i = ps\sum_{j\neq i} v_j + qsv_i \Rightarrow v_i = \frac{ps}{\lambda - qs}\sum_{j\neq i} v_j$. For convenience, let $a = \frac{ps}{\lambda - qs}$. Then, $\sum_i v_i = \sum_i a\sum_{j\neq i} v_j = \sum_i a(k-1)v_i = a(k-1)\sum_i v_i$. Thus, $\sum_i v_i = a(k-1)\sum_i v_i$, and so either $a(k-1) = 1$ or $\sum_i v_i = 0$.

If $a(k-1) = 1$, then $\frac{ps(k-1)}{\lambda - qs} = 1 \Rightarrow \lambda = s(q + p(k-1)) = sq + (n-s)p$. This is just the row sum of every row and corresponds to the eigenvector of all ones. Otherwise, if $\sum_i v_i = 0$, then $v_i = \frac{ps}{\lambda - qs}(-v_i)$. Since v is an eigenvector, v_i is non-zero for some i and this gives $\lambda = (q-p)s$. The eigenspace of $(q-p)s$ consists of the solution space of $\sum_i v_i = 0$ which has dimension k-1. Since A has rank k, the remaining eigenvalues are all 0. Note that if p=0, it is easy to see that every non-zero eigenvalue has value ps. □

Now, we prove a result bounding the deviation of the eigenvalues $\lambda(\hat{A})$ from $\lambda(A)$. Let $\lambda_1, \lambda_2, \lambda_3, ...$ be the eigenvalues of A in order of descending magnitude, and $v_1, v_2, v_3...$ be the corresponding eigenvectors. Similarly, $\hat{\lambda}_j$ and $\hat{v}_j$ for $\hat{A}$. Throughout the paper, $|M|$ will denote the spectral 2-norm of the matrix M, and it is well known that $|M| = \lambda_1(M)$.

Lemma 2. *Let $\hat{A}$ and A be matrices with non-negative eigenvalues. Then, $|\hat{\lambda}_i - \lambda_i| \leq |\hat{A} - A|$*

Proof. By the Courant-Fisher theorem, where x and w_j are n dimensional vectors,

$$\begin{aligned}
\hat{\lambda}_i &= \min_{\substack{\{w_j\} \\ j=1..i-1}} \max_{\substack{|x|=1 \\ (x,w_j)=0}} x^T \hat{A} x \\
&\leq \max_{\substack{|x|=1 \\ (x,v_j)=0}} x^T \hat{A} x \\
&= \max_{\substack{|x|=1 \\ (x,v_j)=0}} x^T (\hat{A} - A) x + \max_{\substack{|x|=1 \\ (x,v_j)=0}} x^T A x \\
&\leq \lambda_1(\hat{A} - A) + \lambda_i
\end{aligned}$$

$\Rightarrow \hat{\lambda}_i - \lambda_i \leq |\hat{A} - A|$.

An analogous proof applied to A instead of $\hat{A}$ gives $\lambda_i - \hat{\lambda}_i \leq |\hat{A} - A|$, and therefore $|\hat{\lambda}_i - \lambda_i| \leq |\hat{A} - A|$. □

The following theorem is a classical result of Furedi and Komlos. See [6] for a proof.

Theorem 3. *Let a_{ij}, $i \geq j$ be independent (not necessarily identically distributed) random variables bounded with a common bound K. Assume that for $i > j$, the a_{ij} have a common expectation μ and variance σ^2. Define a_{ij} for $i < j$ by $a_{ij} = a_{ji}$.*

If $\mu = 0$, then $|A| \leq 2\sigma\sqrt{n} + 50Kn^{\frac{1}{3}} \log n$ with probability at least $1 - \frac{1}{n^{10}}$ for large enough n.

Corollary 4. *$|\hat{\lambda}_i - \lambda_i| \leq 4\sigma\sqrt{n}$ with high probability(whp), for large enough n.*

Proof. We apply Theorem 3 directly to the matrix $\hat{A} - A$, where K=1 to see that $|\hat{A} - A| \leq 2\sigma\sqrt{n} + 50n^{\frac{1}{3}} \log n \leq 4\sigma\sqrt{n}$ with probability at least $1 - \frac{1}{n^{10}}$ for large enough n. Then, from Lemma 3, we obtain $|\hat{\lambda}_i - \lambda_i| \leq |\hat{A} - A| \leq 4\sigma\sqrt{n}$ whp, for large enough n. □

2.2 Proof of Main Theorem

We now proceed to show the clustering capability of the algorithm under this mixture model for t=2. The strategy will be to show that the deviation of $\hat{A}_i^t$ from A_i^t is small relative to the distance between A_i^t and A_j^t, where i and j belong in different blocks. If so, then even after perturbation, rows from different clusters should remain well separated for large enough n. Specifically, if $||\hat{A}_i^t - A_i^t||^2 < \delta$, and $||A_i^t - A_j^t||^2 \geq 16\delta$, then

$$||\hat{A}_{i_1} - \hat{A}_{i_2}||^2 \leq (||\hat{A}_{i_1}^t - A_i^t|| + ||\hat{A}_{i_2}^t - A_i^t||)^2 < 4\delta$$

and

$$\begin{aligned} ||\hat{A}_{i_1} - \hat{A}_{j_1}||^2 &\geq (||A_i - A_j|| - ||\hat{A}_{i_1} - A_i|| - ||\hat{A}_{j_1} - A_j||)^2 \\ &\geq (4\sqrt{\delta} - \sqrt{\delta} - \sqrt{\delta})^2 \\ &= 4\delta \end{aligned}$$

for i_1, i_2 from the cluster of i and j_1 from a different cluster j. Thus, if we choose $\epsilon = ||A_i^t - A_j^t||^2/4 \geq 4\delta$ to be our threshold in the algorithm, then we can cluster correctly in expectation.

First, we present a lemma that shows how block structure is preserved.

Lemma 5. *Let A be a block diagonal matrix with equally sized blocks of size s, with entries of q_a within the blocks, and p_a without. Let B be a matrix with the same block structure and corresponding entries q_b and p_b. Then, AB has the same block structure with corresponding entries $q_{ab} = sq_aq_b + (n-s)p_ap_b$ and $p_{ab} = sq_ap_b + sq_bp_a + (n-2s)p_ap_b$.*

Proof. Let $\Psi(i)$ be the block corresponding to index i. It is clear from $(AB)_{ij} = \sum_k A_{ik}B_{kj} = \sum_k A_{ik}B_{jk}$ that $(AB)_{ij} = sq_aq_b + (n-s)p_ap_b$ when $\Psi(i) = \Psi(j)$, and $(AB)_{ij} = sq_ap_b + sq_bp_a + (n-2s)p_ap_b$ when $\Psi(i) \neq \Psi(j)$. This gives the lemma. □

The following theorem calculates the separation between the rows of A^t from different blocks. In some sense, this is the expected separation between two rows belonging to different clusters.

Theorem 6. $||A_i^t - A_j^t||^2 = 2(q-p)^{2t}(n/k)^{2t-1}$, *where* $\Psi(i) \neq \Psi(j)$

Proof. By Lemma 5, A^t has the same block diagonal structure as A. Let q_t and p_t denote the entries inside and outside of the blocks of A^t, resp., so that $q_1 - p_1 = q - p$. We proceed inductively to show that $q_t - p_t = (q-p)^t s^{t-1}$. By Lemma 5,

$$q_t = sq_{t-1}q + (n-s)p_{t-1}p, \text{ and}$$

$$p_t = sq_{t-1}p + sp_{t-1}q + (n-2s)p_{t-1}p = spq_{t-1} + (sq + (n-2s)p)p_{t-1}$$

$$\begin{aligned} \Rightarrow p_t - q_t &= s(q-p)q_{t-1} + (sp - sq)p_{t-1} \\ &= s(q-p)q_{t-1} - s(q-p)p_{t-1} \\ &= s(q-p)(q_{t-1} - p_{t-1}) \\ &= (q-p)^t s^{t-1} \end{aligned}$$

by the inductive hypothesis.

Thus we know the gap $q_t - p_t$ in general, and this is all we need for the separation: $||A_i^t - A_j^t||^2 = 2s(q_t - p_t)^2 = 2(q-p)^{2t}s^{2t-1} = 2(q-p)^{2t}(n/k)^{2t-1}$.

$\square$

It is easy to see that $||A_i^t - A_j^t||^2 = 0$ when $\Psi(i) = \Psi(j)$, and that $||A_i^t - A_j^t||^2 \geq 2(q-p)^{2t}s_{min}^{2t-1}$, where s_{min} is the size of the smallest block.

The next lemma shows that, relative to the separation between rows from different clusters in A^2, the deviation of $\hat{A}_k^2$ from $\hat{A}_k^2$ is small in expectation. Thus, the "error" from perturbation is bounded.

Lemma 7. $E||\hat{A}_i^2 - A_i^2||^2 \leq 2q^2n^2$

Proof.

$$
\begin{aligned}
E|\hat{A}_i^2 - A_i^2|^2 &= E\left[\sum_m (\sum_l \hat{a}_{il}\hat{a}_{lm} - a_{il}a_{lm})^2\right] \\
&= \sum_m \sum_{l_1,l_2} E\left[(\hat{a}_{il_1}\hat{a}_{l_1m} - a_{il_1}a_{l_1m})(\hat{a}_{il_2}\hat{a}_{l_2m} - a_{il_2}a_{l_2m})\right]
\end{aligned}
$$

the product terms in the expectation are dependent when $l_1 = l_2$

OR if $m \neq i$ and $(l_1, l_2) = (m, i)$ or (i, m)

$$
\begin{aligned}
&= \sum_m \sum_{\substack{l_1\neq l_2 \\ (l_1,l_2)\neq(m,i) \\ (l_1,l_2)\neq(i,m)}} E[\hat{a}_{il_1}\hat{a}_{l_1m} - a_{il_1}a_{l_1m}]E[\hat{a}_{il_2}\hat{a}_{l_2m} - a_{il_2}a_{l_2m}] \\
&\quad + \sum_m \sum_{l_1=l_2} E[(\hat{a}_{il}\hat{a}_{lm} - a_{il}a_{lm})^2] \\
&\quad + \sum_{\substack{m\neq i \\ (l_1,l_2)=(m,i)\text{or}(i,m)}} E[(\hat{a}_{il_1}\hat{a}_{l_1m} - a_{il_1}a_{l_1m})(\hat{a}_{il_2}\hat{a}_{l_2m} - a_{il_2}a_{l_2m})]
\end{aligned}
$$

Note that $\hat{a}_{il} = \hat{a}_{lm}$ only when m=i

$$
\begin{aligned}
&= \sum_{l_1\neq l_2} (a_{il_1} - a_{il_1}^2)(a_{i_l2} - a_{i_l2}^2) + \sum_m \sum_l E[\hat{a}_{il}^2\hat{a}_{lm}^2 - 2a_{il}a_{lm}\hat{a}_{il}\hat{a}_{lm} + a_{il}^2a_{lm}^2] \\
&\quad +2\sum_{m\neq i} E[(\hat{a}_{im}\hat{a}_{mm} - a_{im}a_{mm})(\hat{a}_{ii}\hat{a}_{im} - a_{ii}a_{im})] \\
&= \sum_{l_1,l_2} a_{il_1}(1-a_{il_1})a_{il_2}(1-a_{il_2}) - \sum_l a_{il}^2(1-a_{il})^2 + \sum_m \sum_l E[\hat{a}_{il}^2\hat{a}_{lm}^2] \\
&\quad -2\sum_m \sum_l a_{il}a_{lm}E[\hat{a}_{il}\hat{a}_{lm}] + \sum_m \sum_l a_{il}^2a_{lm}^2 \\
&\quad +2\sum_{m\neq i} a_{ii}a_{mm}a_{im} - 2a_{ii}a_{mm}a_{im}^2 + a_{ii}a_{mm}a_{im}^2
\end{aligned}
$$

We are now in position to expand out all of the expectations.

$$\begin{aligned}
E|\hat{A}_i^2 - A_i^2|^2 &= \left(\sum_l a_{il}(1-a_{il})\right)^2 - \sum_l a_{il}^2(1-a_{il})^2 + \sum_m\sum_l a_{il}a_{lm} - \sum_l a_{il}^2 + \sum_l a_{il} \\
&\quad -2(\sum_m\sum_l a_{il}^2 a_{lm}^2 - \sum_l a_{il}^4 + \sum_l a_{il}^3) + \sum_m\sum_l a_{il}^2 a_{lm}^2 \\
&\quad +2\sum_{m\neq i} a_{ii}a_{mm}a_{im}(1-a_{im}) \\
&= \left(\sum_l a_{il}(1-a_{il})\right)^2 - \sum_l a_{il}^2(1-a_{il})^2 + \sum_m\sum_l a_{il}a_{lm} + \sum_l a_{il}(1-a_{il}) \\
&\quad -\sum_m\sum_l a_{il}^2 a_{lm}^2 - 2\sum_l a_{il}^3(1-a_{il}) \\
&\quad +2[(\sum_m a_{ii}a_{mm}a_{im}(1-a_{im})) - a_{ii}^3(1-a_{ii})] \\
&= (sq(1-q)+(n-s)p(1-p))^2 - (sq^2(1-q)^2 + (n-s)p^2(1-p)^2) \\
&\quad +(sq+(n-s)p)^2 + sq(1-q) + (n-s)p(1-p) - (sq^2+(n-s)p^2)^2 \\
&\quad -2(sq^3(1-q) + (n-s)p^3(1-p)) \\
&\quad +2(q^2(sq(1-q)+(n-s)p(1-p)) - q^3(1-q))
\end{aligned}$$

The n^2 terms dominate, and we may upper bound for n large enough by ignoring the subtracted $(sq^2+(n-s)p^2)^2$ term and the terms linear in n:

$$\begin{aligned}
E|\hat{A}_i^2 - A_i^2|^2 &\leq (sq(1-q)+(n-s)p(1-p))^2 + (sq+(n-s)p)^2 \\
&\leq 2(sq+(n-s)p)^2 \\
&\leq 2q^2n^2
\end{aligned}$$

□

We are now ready to prove the main theorem.

Theorem 8. *For t=2 and some fraction $\delta > 0$, the algorithm correctly clusters $1-\delta$ of the rows with probability at least $\frac{1}{2}$, so long as $|q-p| > \frac{2\sqrt{q}(\frac{k^3}{\delta})^{\frac{1}{4}}}{n^{\frac{1}{4}}}$.*

Proof. Simply let t=2 and $\epsilon = (q-p)^4(n/k)^3/2$. Define a good row to be one for which $||\hat{A}_i^2 - A_i^2|| \leq 4q^2n^2/\delta$. Since $E[\hat{A}_i^2 - A_i^2] \leq 2q^2n^2$ by the above lemma, $Pr[\text{row i is good}] \geq 1-\delta/2$ by applying Markov's Inequality to the bad event. Thus, in expectation, at least $1-\delta/2$ of the rows are good. Again, by a Markov bound applied to the number of bad rows, at least $1-\delta$ of the rows are good with probability at least $\frac{1}{2}$. Now, we see that all the good rows will be classified correctly if $4q^2n^2/\delta < \epsilon/2 = (q-p)^4(n/k)^3/4$, which is equivalent to $|q-p| > \frac{2\sqrt{q}(\frac{k^3}{\delta})^{\frac{1}{4}}}{n^{\frac{1}{4}}}$. Thus, we can correctly cluster $1-\delta$ of the points with probability at least $\frac{1}{2}$ given this probability gap. □

2.3 Optimality of t=2

Unfortunately, further powering of $\hat{A}$ does not improve the clustering. In fact, we show that the gap requirement asymptotically increases due to a rapidly growing error. Specifically, we prove the following lemma:

Lemma 9. *$E[||\hat{A}_i^t - A_i^t||^2] = \Theta(n^{2t-2})$ for all constant $t \geq 2$.*

Proof.

$$\begin{aligned}
E[||\hat{A}_i^t - A_i^t||^2] &= \sum_{j=1}^{n} E\left[(\hat{A}_{ij}^t - A_{ij}^t)^2\right] \\
&= \sum_{j=1}^{n}\left(E\left[(\hat{A}_{ij}^t)^2\right] - 2E\left[\hat{A}_{ij}^t\right]A_{ij}^t + (A_{ij}^t)^2\right) \\
&= \sum_{j=1}^{n} \sum_{k_1,\dots,k_{t-1},k_1',\dots,k_{t-1}' \in [n]^{2t-2}} \Big(E\left[\hat{A}_{ik_1}^t \cdots \hat{A}_{k_{t-1}j}^t \hat{A}_{ik_1'}^t \cdots \hat{A}_{k_{t-1}'j}^t\right] \\
&- 2E\left[\hat{A}_{ik_1}^t \cdots \hat{A}_{k_{t-1}j}^t\right] A_{ik_1'}^t \cdots A_{k_{t-1}'j}^t + A_{ik_1}^t \cdots A_{k_{t-1}j}^t A_{ik_1'}^t \cdots A_{k_{t-1}'j}^t \Big) \\
&\geq \sum_{j=1}^{n} \sum_{k_1,\dots,k_{t-1},k_1',\dots,k_{t-1}' \in [n]^{2t-2}} \Big(E\left[\hat{A}_{ik_1}^t \cdots \hat{A}_{k_{t-1}j}^t \hat{A}_{ik_1'}^t \cdots \hat{A}_{k_{t-1}'j}^t\right] \\
&- E\left[\hat{A}_{ik_1}^t \cdots \hat{A}_{k_{t-1}j}^t\right] A_{ik_1'}^t \cdots A_{k_{t-1}'j}^t \Big)
\end{aligned}$$

Observe that the above expression is a polynomial in n of degree at most $2t-1$. Also, the number of summands in the inner sum for which $|\{k_1, \ldots, k_{t-1}, k'_1, \ldots, k'_{t-1}\}| = l$ is at most $\binom{n}{l} l^{2t-2} = \Theta(n^l)$. Hence, to compute the coefficient of n^{2t-1} in the above it suffices to consider only tuples $(k_1, \ldots, k_{t-1}, k'_1, \ldots, k'_{t-1})$ for which

$$|\{k_1, \ldots, k_{t-1}, k'_1, \ldots, k'_{t-1}\}| = 2t-2$$

In this case though, the expectations split completely so that the inner sum vanishes. It follows that the above is a polynomial in n of degree at most $2t-2$. To compute the coefficient of n^{2t-2}, it suffices to consider tuples $(k_1, \ldots, k_{t-1}, k'_1, \ldots, k'_{t-1})$ for which $|\{k_1, \ldots, k_{t-1}, k'_1, \ldots, k'_{t-1}\}| = 2t-3$, i.e., there is exactly one repetition.

Observe that the inner sum is always positive, so the above is at least:

$$\sum_{j=1}^{n} \sum_{|\{k_1,\ldots,k_{t-1},k'_1,\ldots,k'_{t-1}\}|=2t-3,\ k_1=k'_1} E\left[\hat{A}^t_{ik_1} \cdots \hat{A}^t_{k_{t-1}j} \hat{A}^t_{ik'_1} \cdots \hat{A}^t_{k'_{t-1}j}\right] - E\left[\hat{A}^t_{ik_1} \cdots \hat{A}^t_{k_{t-1}j}\right] A^t_{ik'_1} \cdots A^t_{k'_{t-1}j},$$

which simplifies to

$$\sum_{j=1}^{n} \sum_{|\{k_1,\ldots,k_{t-1},k'_1,\ldots,k'_{t-1}\}|=2t-3,\ k_1=k'_1} (A^t_{ik_1} - (A^t_{ik_1})^2) A^t_{k_1k_2} \cdots A^t_{k_{t-1}j} A^t_{k'_1k'_2} \cdots A^t_{k'_{t-1}j}$$

As long as $p, q > 0$ and $\max\{p(1-p), q(1-q)\} = \Omega(1)$, each term in the inner sum is a positive constant. There are $\binom{n}{2t-3}(2t-3)! = \Theta(n^{2t-3})$ tuples for which $k_1 = k'_1$, so we have $E[||\hat{A}^t_i - A^t_i||^2] = \Omega(n^{2t-2})$, which completes the proof. $\square$

The lemma implies that the gap requirement $q-p$ is $\Omega\left(\frac{1}{n}\right)^{\frac{1}{2t}}$, which is clearly optimal for t=2. This is supported by experimental evidence presented later.

2.4 Blocks of Different Sizes

Here, we justify the earlier claim that it suffices to consider blocks of equal sizes, and that blocks of different sizes do not alter the asymptotics of the performance guarantee by more than constant factors, so long as the minimum block size s_{min} is a constant fraction of n. For the separation, we have previously seen that

$$||A_i^t - A_j^t||^2 \geq 2(q-p)^{2t} s_{min}^{2t-1}$$

It remains to consider the error for unequal blocks. We begin by proving a result about a certain monotonicity property.

Lemma 10. *Let A be the usual symmetric block diagonal matrix of expectations defined previously. Let B be the matrix obtained by symmetrically inserting a row and a column of fractional(probability) entries into A. Then, $E||\hat{A}_i^t - A_i^t||^2 \leq E||\hat{B}_i^t - B_i^t||^2$, where $\hat{A}$ and $\hat{B}$ are the randomized roundings of A and B, resp., preserving symmetry.*

Proof. WLOG and for notational convenience, we may assume that we are inserting the last row and column. Let b_{ij} be the (i,j) entry of B and similarly $\hat{b}_{ij}$ for $\hat{B}$. Define $b = b_{ii_1}b_{i_1i_2}...b_{i_{t-1}j}$, and similarly $\hat{b}$. Analogously define b' and $\hat{b}'$ for the set of indices $i'_1, i'_2, ..., i'_{t-1}$.

$$\begin{aligned}
E||\hat{B}_i^t - B_i^t||^2 &= E\sum_j \left(\sum_{i_1,\ldots,i_{t-1}\in[n+1]} \hat{b}_{ii_1}\hat{b}_{i_1i_2}...\hat{b}_{i_{t-1}j} - b_{ii_1}b_{i_1i_2}...b_{i_{t-1}j} \right)^2 \\
&= E\sum_j \sum_{i_1,\ldots,i_{t-1}\in[n+1]} (\hat{b}-b) \sum_{i'_1,\ldots,i'_{t-1}\in[n+1]} (\hat{b}'-b') \\
&= E\sum_j \sum_{i_1,\ldots,i_{t-1}\in[n+1]} \sum_{i'_1,\ldots,i'_{t-1}\in[n+1]} (\hat{b}-b)(\hat{b}'-b') \\
&= E\sum_j \sum_{i_1,..,i_{t-1}\in[n]} \sum_{i'_1,..,i'_{t-1}\in[n]} (\hat{b}-b)(\hat{b}'-b') + E\sum_j \sum_{i,i'\in S} (\hat{b}-b)(\hat{b}'-b') \\
&= E||\hat{A}_i^t - A_i^t||^2 + E\sum_j \sum_{i,i'\in S} (\hat{b}-b)(\hat{b}'-b')
\end{aligned}$$

where S is the set of tuples for the indices i and i' where at least one index has value n+1. It remains to show that the second summand in the last equation is nonnegative.

$$\begin{aligned} E\sum_j \sum_{i,i' \in S} (\hat{b} - b)(\hat{b}' - b') &= \sum_j \sum_{i,i' \in S} E[\hat{b}\hat{b}' - \hat{b}b' - b\hat{b}' + bb'] \\ &= \sum_j \sum_{i,i' \in S} E[\hat{b}\hat{b}'] - E[\hat{b}b'] - E[b\hat{b}'] + E[bb'] \\ &\geq \sum_j \sum_{i,i' \in S} E[\hat{b}]E[\hat{b}'] - E[\hat{b}]b' - bE[\hat{b}'] + bb' \\ &= \sum_j \sum_{i,i' \in S} (E[\hat{b}] - b)(E[\hat{b}'] - b') \\ &\geq 0. \end{aligned}$$

since $E[\hat{b}\hat{b}'] \geq E[\hat{b}]E[\hat{b}']$ and $E[\hat{b}] - b \geq 0$ for our Bernoulli variables. □

Let A be the original expectations matrix of unequal blocks. Let C be the matrix obtained from A by contracting each block to size s_{min}, and let B be obtained by expanding each block to size s_{max}. Note that we can symmetrically insert rows and columns to obtain B from A, and A from C. From the above lemma, we deduce that the errors increase monotonically:

$$E||\hat{C}_i^t - C_i^t||^2 \leq E||\hat{A}_i^t - A_i^t||^2 \leq E||\hat{B}_i^t - B_i^t||^2$$

We know that the errors for equal sized blocks are polynomials in n of degree $\leq 2t$. Therefore,

$$\left(\frac{s_{max}}{s_{min}}\right)^{2t} E||\hat{C}_i^t - C_i^t||^2 \geq E||\hat{B}_i^t - B_i^t||^2$$

Since s_{min} is a constant fraction of n, $\frac{s_{max}}{s_{min}} = O(1)$. If t is also a constant, then this shows that the error for C is within a constant factor of the error for B, and hence the error for the matrix A of unequal blocks is also within a constant factor of the error for C, the matrix with equal blocks of size s_{min}. This yields the following theorem:

Theorem 11. *Let A be a symmetric block-diagonal matrix of expectations of unequal* *blocks, where s_{min} is a constant fraction of n. Let C be obtained from A by contracting each block to size s_{min}. Then, for some constant r depending on the constant t,*

$$E||\hat{A}_i^t - A_i^t||^2 \leq rE||\hat{C}_i^t - C_i^t||^2$$

From this theorem, we may conclude that the asymptotics of the performance guarantees are unaffected by taking unequal blocks, and that the algorithm continues to work in this more general setting, for constant values of t.

Chapter 3

Experiments

The experimental evidence is encouraging and indicates some interesting behavior. First, we would like to examine the performance of the algorithm for different values of the power t. We generate the matrix $\hat{A}$ from the matrix A via randomized rounding preserving symmetry as specified by the mixture model with q=0.45, p=0.05, and N=200 nodes divided evenly into 4 clusters. The success of the algorithm is measured by the percentage of the $\binom{N}{2}$ pairwise relationships(classified as same cluster or different) that it guesses correctly. Note that a score of 75% is not impressive and corresponds to the case where every node is classified to its own cluster. In the other extreme, a score of 25% corresponds to the case in which all of the nodes are classified to the same cluster. The results are shown in figure 3-1, percentage correct against t.

Notice that the results basically conform to theoretical expectations, but the algorithm seems to perform unusually well for t=3. We find this to be purely a matter of constant factors, as the power of q in the leading coefficient for the error of t=3 is larger than the corresponding power for t=2. Were q in our experiment much closer to 1 than 0.45, this effect would not be observed.

In addition, we would like to see how performance varies with probability gap, and to verify our intuition that clustering should become easier with larger gaps. We again instantiate the mixture model with N=200 nodes divided evenly into 4 clusters

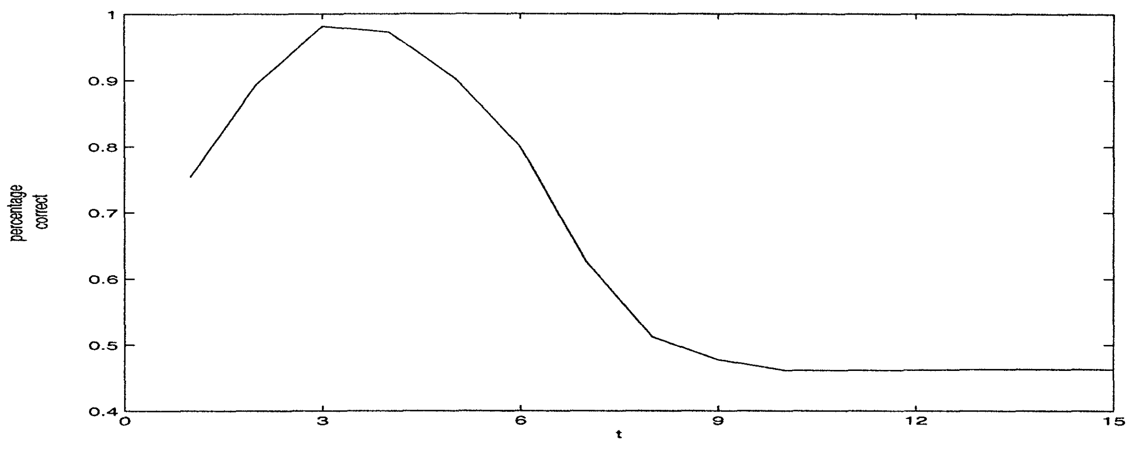

Figure 3-1: percentage correct vs t

and p=0.05. We plot the percentage correct for t=3 against varying probability gaps in figure 3-2.

In this paper, we provide theoretical guarantees for the case of t=2. We show that this is in fact the optimal case, and that the performance degrades with larger values of t. The algorithm is inherently weaker and does not achieve McSherry's bound of $\Theta(\frac{1}{\sqrt{n}})$. However, our algorithm is elegant and simple, compared to the more complicated SVD computation and random splitting used in McSherry's procedure. Furthermore, matrix exponentiation runs in $O(n^{2.37})$ time using the theoretically best algorithm, and in $O(n^{2.7})$ time using the more practical and often used Strassen's algorithm. This is significantly faster than the $O(n^3)$ time required to compute the SVD and thus our algorithm should be well suited to large data sets where the gap requirement of $\Omega(\frac{1}{n^{\frac{1}{4}}})$ is easily satisfied and running time is a major consideration.

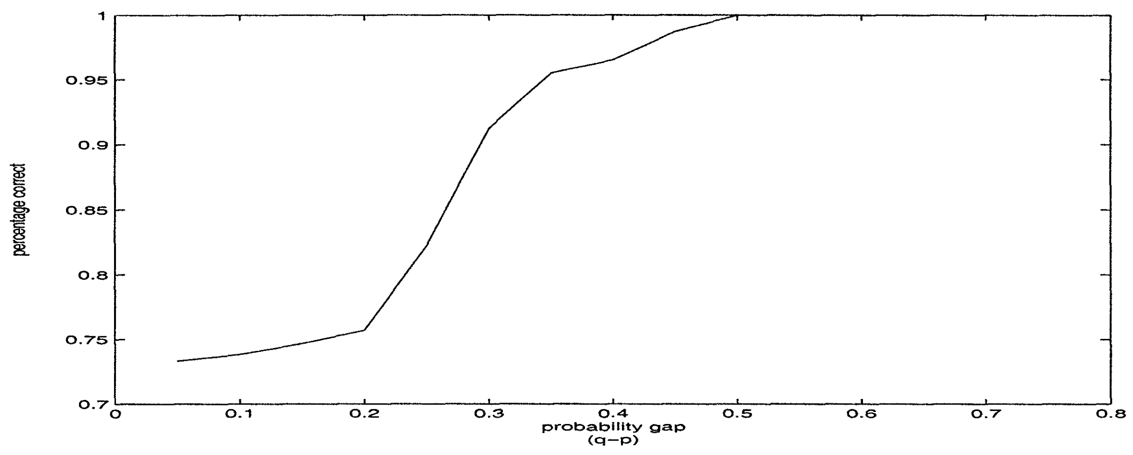

Figure 3-2: percentage correct vs gap

Bibliography

[1] Yossi Azar, Amos Fiat, Anna Karlin, and Frank McSherry. Data mining through spectral analysis. In *IEEE Symposium on Foundations of Computer Science*, 2001.

[2] Avrim Blum and Joel Spencer. Coloring random and semi-random k-colorable graphs. In *Journal of Algorithms*, 1995.

[3] Ravi Boppana. Eigenvalues and graph bisection: An average-case analysis. In *IEEE Symposium on Foundations of Computer Science*, pages 280–285, 1985.

[4] Anne Condon and Richard Karp. Algorithms for graph partitioning on the planted partition model. In *Random Structure and Algorithms*, 1999.

[5] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *ACM-SIAM Symposium on Discrete Algorithms*, 1999.

[6] Zoltan Furedi and Janos Komlos. The eigenvalues of random symmetric matrices. In *Combinatorica*, pages 233–241, 1981.

[7] Gene Golub and Charles Van Loan. In *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.

[8] Mark Jerrum and Gregory Sorkin. Simulated annealing for graph bisection. In *IEEE Symposium on Foundations of Computer Science*, 1993.

[9] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings: Good, bad, and spectral. In *IEEE Symposium on Foundations of Computer Science*, 2000.

[10] Michael Krivelevich and Van Vu. On the concentration of eigenvalues of random symmetric matrices. Technical Report 60, Microsoft, 2000.

[11] Laszlo Lovasz. Random walks on graphs: a survey. In *Combinatorics*, pages 1–46, 1993.

[12] Frank McSherry. Spectral partitioning of random graphs. In *IEEE Symposium on Foundations of Computer Science*, 2001.

[13] Christos Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: a probabilistic analysis. In *ACM Conference on Principles of Database Systems*, 1998.

[14] G.W. Stewart. In *Introduction to Matrix Computations.* Academic Press, 1973.

[15] Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. In *IEEE Symposium on Foundations of Computer Science*, 2002.