Spectral Algorithms for Data Analysis (draft)

Ravindran Kannan and Santosh S. Vempala

September 27, 2021

Summary. Spectral methods refer to the use of eigenvalues, eigenvectors, singular values and singular vectors. They are widely used in Engineering, Applied Mathematics and Statistics. More recently, spectral methods have found numerous applications in Computer Science to "discrete" as well "continuous" problems. This book describes modern applications of spectral methods, and novel algorithms for estimating spectral parameters.

In the first part of the book, we present applications of spectral methods to problems from a variety of topics including combinatorial optimization, learning and clustering.

The second part of the book is motivated by efficiency considerations. A feature of many modern applications is the massive amount of input data. While sophisticated algorithms for matrix computations have been developed over a century, a more recent development is algorithms based on "sampling on the fly" from massive matrices. Good estimates of singular values and low rank approximations of the whole matrix can be provably derived from a sample. Our main emphasis in the second part of the book is to present these sampling methods with rigorous error bounds. We also present recent extensions of spectral methods from matrices to tensors and their applications to some combinatorial optimization problems.

Contents

Part I

Applications

Chapter 1

The Best-Fit Subspace

To provide an in-depth and relatively quick introduction to SVD and its applicability, in this opening chapter, we consider the *best-fit subspace* problem. Finding the best-fit line for a set of data points is a classical problem. A natural measure of the quality of a line is the least squares measure, the sum of squared (perpendicular) distances of the points to the line. A more general problem, for a set of data points in $\mathbf{R}^n$, is finding the best-fit k-dimensional subspace. SVD can be used to find a subspace that minimizes the sum of squared distances to the given set of points in polynomial time. In contrast, for other measures such as the sum of distances or the maximum distance, no polynomial-time algorithms are known.

A clustering problem widely studied in theoretical computer science is the k-means problem. The goal is to find a set of k points that minimize the sum of their squared distances of the data points to their nearest facilities. A natural relaxation of the k-means problem is to find the k-dimensional subspace for which the sum of the distances of the data points to the subspace is minimized (we will see that this is a relaxation). We will apply SVD to solve this relaxed problem and use the solution to approximately solve the original problem.

1.1 Singular Value Decomposition

For an $n \times n$ matrix A, an eigenvalue λ and corresponding eigenvector v satisfy the equation

$$Av = \lambda v.$$

In general, i.e., if the matrix has nonzero determinant, it will have n nonzero eigenvalues (not necessarily distinct). For an introduction to the theory of eigenvalues and eigenvectors, several textbooks are available.

Here we deal with an $m \times n$ rectangular matrix A, where the m rows denoted $A_{(1)}, A_{(2)}, \ldots A_{(m)}$ are points in $\mathbf{R}^n$; $A_{(i)}$ will be a row vector.

If $m \neq n$, the notion of an eigenvalue or eigenvector does not make sense, since the vectors Av and λv have different dimensions. Instead, a *singular value*

σ and corresponding *singular vectors* $u \in \boldsymbol{R}^m, v \in \boldsymbol{R}^n$ simultaneously satisfy the following two equations

1. $Av = \sigma u$
2. $u^T A = \sigma v^T$.

We can assume, without loss of generality, that u and v are unit vectors. To see this, note that a pair of singular vectors u and v must have equal length, since $u^T Av = \sigma \|u\|^2 = \sigma \|v\|^2$. If this length is not 1, we can rescale both by the same factor without violating the above equations.

Now we turn our attention to the value $\max_{\|v\|=1} \|Av\|^2$. Since the rows of A form a set of m vectors in R^n, the vector Av is a list of the projections of these vectors onto the line spanned by v, and $\|Av\|^2$ is simply the sum of the squares of those projections.

Instead of choosing v to maximize $\|Av\|^2$, the Pythagorean theorem allows us to equivalently choose v to minimize the sum of the squared distances of the points to the line through v. In this sense, v defines the line through the origin that best fits the points.

To argue this more formally, Let $d(A_{(i)}, v)$ denote the distance of the point $A_{(i)}$ to the line through v. Alternatively, we can write

$$d(A_{(i)}, v) = \|A_{(i)} - (A_{(i)}v)v^T\|.$$

For a unit vector v, the Pythagorean theorem tells us that

$$\|A_{(i)}\|^2 = \|(A_{(i)}v)v^T\|^2 + d(A_{(i)}, v)^2.$$

Thus we get the following proposition. Note that $\|A\|_F^2 = \sum_{i,j} A_{ij}^2$ refers to the squared Frobenius norm of A.

Proposition 1.1.

$$\max_{\|v\|=1} \|Av\|^2 = ||A||_F^2 - \min_{\|v\|=1} \|A-(Av)v^T\|_F^2 = ||A||_F^2 - \min_{\|v\|=1} \sum_i \|A_{(i)}-(A_{(i)}v)v^T\|^2$$

Proof. We simply use the identity:

$$\|Av\|^2 = \sum_i \|(A_{(i)}v)v^T\|^2 = \sum_i \|A_{(i)}\|^2 - \sum_i \|A_{(i)} - (A_{(i)}v)v^T\|^2$$

□

The proposition says that the v which maximizes $\|Av\|^2$ is the "best-fit" vector which also minimizes $\sum_i d(A_{(i)}, v)^2$.

Next, we claim that v is in fact a singular vector.

Proposition 1.2. *The vector $v_1 = \arg\max_{\|v\|=1} \|Av\|^2$ is a singular vector, and moreover $\|Av_1\|$ is the largest (or "top") singular value.*

Proof. For any singular vector v,

$$(A^TA)v = \sigma A^Tu = \sigma^2 v.$$

Thus, v is an eigenvector of A^TA with corresponding eigenvalue σ^2. Conversely, an eigenvector of A^TA is also a singular vector of A. To see this, let v be an eigenvector of A^TA with corresponding eigenvalue λ. Note that λ is positive, since

$$\|Av\|^2 = v^TA^TAv = \lambda v^Tv = \lambda\|v\|^2$$

and thus

$$\lambda = \frac{\|Av\|^2}{\|v\|^2}.$$

Now if we let $\sigma = \sqrt{\lambda}$ and $u = Av/\sigma$, it is easy to verify that u,v, and σ satisfy the singular value requirements. The right singular vectors $\{v_i\}$ are thus eigenvectors of A^TA.

Now we can also write

$$\|Av\|^2 = v^T(A^TA)v.$$

Viewing this as a function of v, $f(v) = v^T(A^TA)v$, its gradient is

$$\nabla f(v) = 2(A^TA)v.$$

Thus, any *local* maximum of this function on the unit sphere must satisfy

$$\nabla f(v) = \lambda v$$

for some λ, i.e., $A^TAv = \lambda v$ for some scalar λ. So any local maximum is an eigenvector of A^TA. Since v_1 is a global maximum of f, it must also be a local maximum and therefore an eigenvector of A^TA.

□

More generally, we consider a k-dimensional subspace that best fits the data. It turns out that this space is specified by the top k singular vectors, as stated precisely in the following proposition.

Theorem 1.3. *Define the k-dimensional subspace V_k as the span of the following k vectors:*

$$\begin{aligned}
v_1 &= \arg\max_{\|v\|=1} \|Av\| \\
v_2 &= \arg\max_{\|v\|=1, v\cdot v_1=0} \|Av\| \\
&\vdots \\
v_k &= \arg\max_{\|v\|=1, v\cdot v_i=0 \ \forall i<k} \|Av\|,
\end{aligned}$$

where ties for any arg max *are broken arbitrarily. Then* V_k *is* optimal *in the sense that*

$$V_k = \arg \min_{dim(V)=k} \sum_i d(A_{(i)}, V)^2.$$

Further, $v_1, v_2, ..., v_n$ *are all singular vectors, with corresponding singular values* $\sigma_1, \sigma_2, ..., \sigma_n$ *and*

$$\sigma_1 = \|Av_1\| \geq \sigma_2 = \|Av_2\| \geq ... \geq \sigma_n = \|Av_n\|.$$

Finally, $A = \sum_{i=1}^n \sigma_i u_i v_i^T$.

Such a decomposition where,

1. The sequence of σ_i's is nonincreasing
2. The sets $\{u_i\}, \{v_i\}$ are orthonormal

is called the *Singular Value Decomposition (SVD)* of A.

Proof. We first prove that V_k are optimal by induction on k. The case $k = 1$ is by definition. Assume that V_{k-1} is optimal.

Suppose V_k' is an optimal subspace of dimension k. Then we can choose an orthonormal basis for V_k', say $w_1, w_2, \ldots w_k$, such that w_k is orthogonal to V_{k-1}. By the definition of V_k', we have that

$$||Aw_1||^2 + ||Aw_2^2|| + \ldots ||Aw_k||^2$$

is maximized (among all sets of k orthonormal vectors.) If we replace w_i by v_i for $i = 1, 2, \ldots, k-1$, we have

$$\|Aw_1\|^2 + \|Aw_2^2\| + \ldots \|Aw_k\|^2 \leq \|Av_1\|^2 + \ldots + \|Av_{k-1}\|^2 + \|Aw_k\|^2.$$

Therefore we can assume that V_k' is the span of V_{k-1} and w_k. It then follows that $\|Aw_k\|^2$ maximizes $\|Ax\|^2$ over all unit vectors x orthogonal to V_{k-1}.

Proposition 1.2 can be extended to show that $v_1, v_2, ..., v_n$ are all singular vectors. The assertion that $\sigma_1 \geq \sigma_2 \geq .... \geq \sigma_n \geq 0$ follows from the definition of the v_i's.

We can verify that the decomposition

$$A = \sum_{i=1}^n \sigma_i u_i v_i^T$$

is accurate. This is because the vectors $v_1, v_2, ..., v_n$ form an orthonormal basis for $\boldsymbol{R}^n$, and the action of A on any v_i is equivalent to the action of $\sum_{i=1}^n \sigma_i u_i v_i^T$ on v_i. □

Note that we could actually decompose A into the form $\sum_{i=1}^n \sigma_i u_i v_i^T$ by picking $\{v_i\}$ to be any orthogonal basis of $\boldsymbol{R}_n$, but the proposition actually

states something stronger: that we can pick $\{v_i\}$ in such a way that $\{u_i\}$ is also an orthogonal set.

We state one more classical theorem. We have seen that the span of the top k singular vectors is the best-fit k-dimensional subspace for the rows of A. Along the same lines, the partial decomposition of A obtained by using only the top k singular vectors is the best rank-k matrix approximation to A.

Theorem 1.4. *Among all rank k matrices D, the matrix $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$ is the one which minimizes $\|A - D\|_F^2 = \sum_{i,j}(A_{ij} - D_{ij})^2$. Further,*

$$\|A - A_k\|_F^2 = \sum_{i=k+1}^{n} \sigma_i^2.$$

Proof. We have

$$\|A - D\|_F^2 = \sum_{i=1}^{m} \|A_{(i)} - D_{(i)}\|^2.$$

Since D is of rank at most k, we can assume that all the $D_{(i)}$ are projections of $A_{(i)}$ to some rank k subspace and therefore,

$$\begin{aligned} \sum_{i=1}^{m} \|A_{(i)} - D_{(i)}\|^2 &= \sum_{i=1}^{m} \|A_{(i)}\|^2 - \|D_{(i)}\|^2 \\ &= \|A\|_F^2 - \sum_{i=1}^{m} \|D_{(i)}\|^2. \end{aligned}$$

Thus the subspace is exactly the SVD subspace given by the span of the first k singular vectors of A. □

1.2 Algorithms for computing the SVD

Computing the SVD is a major topic of numerical analysis [Str88, GvL96, Wil88]. Here we describe a basic algorithm called the power method.

Assume that A is symmetric.

1. Let x be a random unit vector.
2. Repeat:
$$x := \frac{Ax}{\|Ax\|}$$

For a nonsymmetric matrix A, we can simply apply the power iteration to $A^T A$.

Exercise 1.1. *Show that with probability at least $1/4$, the power iteration applied k times to a symmetric matrix A finds a vector x^k such that*

$$\|Ax^k\|^2 \geq \left(\frac{1}{4n}\right)^{1/k} \sigma_1^2(A).$$

[Hint: First show that $\|Ax^k\| \geq (|x \cdot v|)^{1/k}\sigma_1(A)$ *where* x *is the starting vector and* v *is the top eigenvector of* A*; then show that for a random unit vector* x*, the random variable* $|x.v|$ *is large with some constant probability].*

The second part of this book deals with faster, sampling-based algorithms.

1.3 The k-means clustering problem

This section contains a description of a clustering problem which is often called k-means in the literature and can be solved approximately using SVD. This illustrates a typical use of SVD and has a provable bound.

We are given m points $\mathcal{A} = \{A^{(1)}, A^{(2)}, \ldots A^{(m)}\}$ in n-dimensional Euclidean space and a positive integer k. The problem is to find k points $\mathcal{B} = \{B^{(1)}, B^{(2)}, \ldots, B^{(k)}\}$ such that

$$f_{\mathcal{A}}(\mathcal{B}) = \sum_{i=1}^{m} (\text{dist}(A^{(i)}, \mathcal{B}))^2$$

is minimized. Here $\text{dist}(A^{(i)}, \mathcal{B})$ is the Euclidean distance of $A^{(i)}$ to its nearest point in $\mathcal{B}$. Thus, in this problem we wish to minimize the sum of squared distances to the nearest "cluster center". This is commonly called the k-means or k-means clustering problem. It is NP-hard even for $k = 2$. A popular local search heuristic for this problem is often called the k-means algorithm.

We first observe that the solution is given by k clusters S_j, $j = 1, 2, \ldots k$. The cluster center $B^{(j)}$ will be the centroid of the points in S_j, $j = 1, 2, \ldots, k$. This is seen from the fact that for any set $\mathcal{S} = \{X^{(1)}, X^{(2)}, \ldots, X^{(r)}\}$ and any point B we have

$$\sum_{i=1}^{r} \|X^{(i)} - B\|^2 = \sum_{i=1}^{r} \|X^{(i)} - \bar{X}\|^2 + r\|B - \bar{X}\|^2, \tag{1.1}$$

where $\bar{X}$ is the centroid $(X^{(1)} + X^{(2)} + \cdots + X^{(r)})/r$ of $\mathcal{S}$. The next exercise makes this clear.

Exercise 1.2. *Show that for a set of point* $X^1, \ldots, X^k \in \boldsymbol{R}^n$*, the point* Y *that minimizes* $\sum_{i=1}^{k} |X^i - Y|^2$ *is their centroid. Give an example when the centroid is not the optimal choice if we minimize sum of distances rather than squared distances.*

The k-means clustering problem is thus the problem of partitioning a set of points into clusters so that the *sum of the squared distances to the means*, i.e., the variances of the clusters is minimized.

We define a relaxation of this problem that we may cal the *Continuous Clustering Problem* (CCP): find the subspace V of $\mathbf{R}^n$ of dimension at most k that minimizes

$$g_{\mathcal{A}}(V) = \sum_{i=1}^{m} \text{dist}(A^{(i)}, V)^2.$$

The reader will recognize that this can be solved using the SVD. It is easy to see that the optimal value of the k-means clustering problem is an upper bound for the optimal value of the CCP. Indeed for any set $\mathcal{B}$ of k points,

$$f_{\mathcal{A}}(\mathcal{B}) \geq g_{\mathcal{A}}(V_{\mathcal{B}}) \tag{1.2}$$

where $V_{\mathcal{B}}$ is the subspace generated by the points in $\mathcal{B}$.

We now present a factor 2 approximation algorithm for the k-means clustering problem using the relaxation to the best-fit subspace. The algorithm has two parts. First we project to the k-dimensional SVD subspace, solving the CCP. Then we solve the problem in the low-dimensional space using a brute-force algorithm with the following guarantee.

Theorem 1.5. *The k-means problem can be solved in $O(m^{k^2 d/2})$ time when the input $\mathcal{A} \subseteq \mathbf{R}^d$.*

We describe the algorithm for the low-dimensional setting. Each set $\mathcal{B}$ of "cluster centers" defines a Voronoi diagram where cell $\mathcal{C}_i = \{X \in \mathbf{R}^d : |X - B^{(i)}| \leq |X - B^{(j)}| \text{ for } j \neq i\}$ consists of those points whose closest point in $\mathcal{B}$ is $B^{(i)}$. Each cell is a polyhedron and the total number of faces in $C_1, C_2, \ldots, C_k$ is no more than $\binom{k}{2}$ since each face is the set of points equidistant from two points of $\mathcal{B}$.

We have seen in (1.1) that it is the partition of $\mathcal{A}$ that determines the best $\mathcal{B}$ (via computation of centroids) and so we can move the boundary hyperplanes of the optimal Voronoi diagram, without any face passing through a point of $\mathcal{A}$, so that each face contains at least d points of $\mathcal{A}$.

Assume that the points of $\mathcal{A}$ are in general position and $0 \notin \mathcal{A}$ (a simple perturbation argument deals with the general case). This means that each face now contains d affinely independent points of $\mathcal{A}$. We ignore the information about which side of each face to place these points and so we must try all possibilities for each face. This leads to the following enumerative procedure for solving the k-means clustering problem:

Algorithm: Voronoi-k-means

1. Enumerate all sets of t hyperplanes, such that $k \leq t \leq k(k-1)/2$ hyperplanes, and each hyperplane contains d affinely independent points of $\mathcal{A}$. The number of sets is at most
$$\sum_{t=k}^{\binom{k}{2}} \binom{\binom{m}{d}}{t} = O(m^{dk^2/2}).$$
2. Check that the arrangement defined by these hyperplanes has exactly k cells.
3. Make one of 2^{td} choices as to which cell to assign each point of $\mathcal{A}$ which lies on a hyperplane
4. This defines a unique partition of $\mathcal{A}$. Find the centroid of each set in the partition and compute $f_{\mathcal{A}}$.

Now we are ready for the complete algorithm. As remarked previously, CCP can be solved by Linear Algebra. Indeed, let V be a k-dimensional subspace of $\mathbf{R}^n$ and $\bar{A}^{(1)}, \bar{A}^{(2)}, \ldots, \bar{A}^{(m)}$ be the orthogonal projections of $A^{(1)}, A^{(2)}, \ldots, A^{(m)}$ onto V. Let $\bar{A}$ be the $m \times n$ matrix with rows $\bar{A}^{(1)}, \bar{A}^{(2)}, \ldots, \bar{A}^{(m)}$. Thus $\bar{A}$ has rank at most k and

$$\|A - \bar{A}\|_F^2 = \sum_{i=1}^{m} |A^{(i)} - \bar{A}^{(i)}|^2 = \sum_{i=1}^{m} (\text{dist}(A^{(i)}, V))^2.$$

Thus to solve CCP, all we have to do is find the first k vectors of the SVD of A (since by Theorem (1.4), these minimize $\|A - \bar{\mathbf{A}}\|_F^2$ over all rank k matrices $\bar{A}$) and take the space V_{SVD} spanned by the first k singular vectors in the row space of A.

We now show that combining SVD with the above algorithm gives a 2-approximation to the k-means problem in arbitrary dimension. Let $\bar{\mathcal{A}} = \{\bar{A}^{(1)}, \bar{A}^{(2)}, \ldots, \bar{A}^{(m)}\}$ be the projection of $\mathcal{A}$ onto the subspace V_k. Let $\bar{\mathcal{B}} = \{\bar{B}^{(1)}, \bar{B}^{(2)}, \ldots, \bar{B}^{(k)}\}$ be the optimal solution to k-means problem with input $\bar{\mathcal{A}}$.

Algorithm for the k-means clustering problem

- Compute V_k.
- Solve the k-means clustering problem with input $\bar{\mathcal{A}}$ to obtain $\bar{\mathcal{B}}$.
- Output $\bar{\mathcal{B}}$.

It follows from (1.2) that the optimal value $Z_{\mathcal{A}}$ of the k-means clustering problem satisfies

$$Z_{\mathcal{A}} \geq \sum_{i=1}^{m} |A^{(i)} - \bar{A}^{(i)}|^2. \tag{1.3}$$

Note also that if $\hat{\mathcal{B}} = \{\hat{B}^{(1)}, \hat{B}^{(2)}, \ldots, \hat{B}^{(k)}\}$ is an optimal solution to the k-means clustering problem and $\tilde{\mathcal{B}}$ consists of the projection of the points in $\hat{\mathcal{B}}$ onto V, then

$$Z_{\mathcal{A}} = \sum_{i=1}^{m} \text{dist}(A^{(i)}, \hat{\mathcal{B}})^2 \geq \sum_{i=1}^{m} \text{dist}(\bar{A}^{(i)}, \tilde{\mathcal{B}})^2 \geq \sum_{i=1}^{m} \text{dist}(\bar{A}^{(i)}, \bar{\mathcal{B}})^2.$$

Combining this with (1.3) we get

$$\begin{aligned} 2Z_{\mathcal{A}} &\geq \sum_{i=1}^{m} (|A^{(i)} - \bar{A}^{(i)}|^2 + \text{dist}(\bar{A}^{(i)}, \bar{\mathcal{B}})^2) \\ &= \sum_{i=1}^{m} \text{dist}(A^{(i)}, \bar{\mathcal{B}})^2 \\ &= f_{\mathcal{A}}(\bar{\mathcal{B}}) \end{aligned}$$

proving that we do indeed get a 2-approximation.

Theorem 1.6. *The above algorithm for the k-means clustering problem finds a factor 2 approximation for m points in $\boldsymbol{R}^n$ in $O(mn^2 + m^{k^3/2})$ time.*

1.4 Discussion

In this chapter, we reviewed basic concepts in linear algebra from a geometric perspective. The k-means problem is a typical example of how SVD is used: project to the SVD subspace, then solve the original problem. In many application areas, the method known as "Principal Component Analysis" (PCA) uses the projection of a data matrix to the span of the largest singular vectors. There are several introducing the theory of eigenvalues and eigenvectors as well as SVD/PCA, e.g., [GvL96, Str88, Bha97].

The application of SVD to the k-means clustering problem is from [DFK$^+$04] and its hardness is from [ADHP09]. The following complexity questions are open: (1) Given a matrix A, is it NP-hard to find a rank-k matrix D that minimizes the error with respect to the L_1 norm, i.e., $\sum_{i,j} |A_{ij} - D_{ij}|$? (more generally for L_p norm for $p \neq 2$)? (2) Given a set of m points in $\boldsymbol{R}^n$, is it NP-hard to find a subspace of dimension at most k that minimizes the sum of distances of the points to the subspace? It is known that finding a subspace that minimizes the maximum distance is NP-hard [MT82]; see also [HPV02].

Chapter 2

Unraveling Mixtures Models

An important class of data models are generative, i.e., they assume that data is generated according to a probability distribution D in $\boldsymbol{R}^n$. One major scenario is when D is a mixture of some special distributions. These may be continuous or discrete. Prominent and well-studied instances of each are:

- D is a *mixture* of Gaussians.
- D is choosing the row vectors of the adjacency matrix of a *random graph* with certain special properties.

For the second situation, we will see in Chapter 4 that spectral methods are quite useful. In this chapter, we study a classical generative model where the input is a set of points in $\boldsymbol{R}^n$ drawn randomly from a mixture of probability distributions. The sample points are unlabeled and the basic problem is to correctly classify them according the component distribution which generated them. The special case when the component distributions are Gaussians is a classical problem and has been widely studied. In later chapters, we will revisit mixture models in other guises (e.g., random planted partitions).

Let F be a probability distribution in $\boldsymbol{R}^n$ with the property that it is a convex combination of distributions of known type, i.e., we can decompose F as

$$F = w_1F_1 + w_2F_2 + \cdots + w_kF_k$$

where each F_i is a probability distribution with mixing weight $w_i \geq 0$, and $\sum_i w_i = 1$. A random point from F is drawn from distribution F_i with probability w_i.

Given a sample of points from F, we consider the following problems:

1. Classify the sample according to the component distributions.
2. Learn parameters of the component distributions (e.g., estimate their means, covariances and mixing weights).

The second problem is well-defined by the following theorem.

Theorem 2.1. *A mixture of Gaussians can be uniquely determined by its probability density function.*

For most of this chapter, we deal with the classical setting: each F_i is a Gaussian in $\boldsymbol{R}^n$. In fact, we begin with the special case of spherical Gaussians whose density functions (i) depend only on the distance of a point from the mean and (ii) can be written as the product of density functions on each coordinate. The density function of a spherical Gaussian in $\boldsymbol{R}^n$ is

$$p(x) = \frac{1}{(\sqrt{2\pi}\sigma)^n} e^{-\|x-\mu\|^2/2\sigma^2}$$

where μ is its mean and σ is the standard deviation along any direction.

2.1 The challenge of high dimensionality

Intuitively, if the component distributions are far apart, so that a pair of points from the same component distribution are closer to each other than any pair from different components, then classification is straightforward. If the component distributions have a large overlap, then it is not possible to correctly classify all or most of the points, since points from the overlap could belong to more than one component. To illustrate this, consider a mixture of two one-dimensional Gaussians with means μ_1, μ_2 and variances σ_1, σ_2. For the overlap of the distributions to be smaller than ϵ, we need the means to be separated as

$$|\mu_1 - \mu_2| \geq C\sqrt{\log(1/\epsilon)}\max\{\sigma_1, \sigma_2\}.$$

$|\mu_1 - \mu_2| \geq C\sqrt{\log(1/\epsilon)}\min\{\sigma_1, \sigma_2\}$. If the distance were smaller than this by a constant factor, then the total variation (or L_1) distance between the two distributions would be less than $1 - \epsilon$ and we could not correctly classify with high probability a $1-\epsilon$ fraction of the mixture. On the other hand, if the means were separated as above, for a sufficiently large C, then at least $1 - \epsilon$ of the sample can be correctly classified with high probability; if we replace $\sqrt{\log(1/\epsilon)}$ with $\sqrt{\log m}$ where m is the size of the sample, then with high probability, every pair of points from different components would be farther apart than any pair of points from the same component, and classification is easy. For example, we can use the following distance-based classification algorithm (sometimes called *single linkage*):

1. Sort all pairwise distances in increasing order.
2. Choose edges in this order till the edges chosen form exactly two connected components.
3. Declare points in each connected component to be from the same component distribution of the mixture.

Now consider a mixture of two spherical Gaussians, but in $\boldsymbol{R}^n$. We claim that the same separation as above with distance between the means measured as Euclidean length, suffices to ensure that the components are probabilistically separated. Indeed, this is easy to see by considering the projection of the mixture to the line joining the two original means. The projection is a mixture of two one-dimensional Gaussians satisfying the required separation condition above. Will the above classification algorithm work with this separation? The answer turns out to be no. This is because in high dimension, the distances between pairs from different components, although higher in expectation compared to distances from the same component, can deviate from their expectation by factors that depend both on the variance *and the ambient dimension*, and so, the separation required for such distance-based methods to work grows as a function of the dimension. We will discuss this difficulty and how to get around in more detail presently.

The classification problem is inherently tied to the mixture being separable. However, the learning problem, in principle, does not require separable mixtures. In other words, one could formulate the problem of estimating the parameters of the mixture without assuming any separation between the components. For this learning problem with no separation, even for mixtures of Gaussians, there is an exponential lower bound in k, the number of components, on the time and sample complexity. Most of this chapter is about polynomial algorithms for the classification and learning problems under suitable assumptions.

2.2 Classifying separable mixtures

In order to correctly identify sample points, we require the overlap of distributions to be small . How can we quantify the distance between distributions? One way, if we only have two distributions, is to take the total variation distance,

$$d_{TV}(f_1, f_2) = \frac{1}{2} \int_{\boldsymbol{R}^n} |f_1(x) - f_2(x)| \, dx,$$

where f_1, f_2 are density functions of the two distributions. The overlap of two distributions is defined as $1 - d_{TV}(f_1, f_2)$. We can require this to be large for two well-separated distributions, i.e., $d_{TV}(f_1, f_2) \geq 1 - \epsilon$, if we tolerate ϵ error.

This can be generalized in two ways to $k > 2$ components. First, we could require the above condition holds for every pair of components, i.e., pairwise probabilistic separation. Or we could have the following single condition.

$$\int_{\boldsymbol{R}^n} \left(2 \max_i w_i f_i(x) - \sum_{i=1}^{k} w_i f_i(x) \right)^+ dx \geq 1 - \epsilon \tag{2.1}$$

where

$$x^+ = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

The quantity inside the integral is simply the maximum $w_i f_i$ at x, minus the sum of the rest of the $w_i f_i$'s. If the supports of the components are essentially disjoint, the integral will be 1.

For $k > 2$, it is not known how to efficiently classify mixtures when we are given one of these probabilistic separations. In what follows, we use stronger assumptions. Strengthening probabilistic separation to geometric separation turns out to be quite effective. We consider that next.

Geometric separation. For two distributions, we require $\|\mu_1 - \mu_2\|$ to be large compared to $\max\{\sigma_1, \sigma_2\}$. Note this is a stronger assumption than that of small overlap. In fact, two distributions can have the *same* mean, yet still have small overlap, e.g., two spherical Gaussians with different variances. In one dimension, probabilistic separation implies either mean separation or variance separation.

Theorem 2.2. *Suppose $F_1 = N(\mu_1, 1)$ and $F_2 = N(\mu_2, \sigma^2)$ are two 1-dimensional Gaussians. If $d_{TV}(F_1, F_2) \geq 1 - \epsilon$, then either $|\mu_1 - \mu_2|^2 \geq \log(1/\epsilon) - 1$ or $\max\{\sigma^2, 1/\sigma^2\} \geq \log(1/\epsilon)$.*

Proof. The KL-divergence between F_1 and F_2 has a closed form:

$$\mathrm{KL}(F_1 \parallel F_2) = \frac{1}{2}\left((\sigma^2 - 1) + (\mu_1 - \mu_2)^2 - \log \sigma^2\right).$$

By Vajda's lower bound, we have

$$\mathrm{KL}(F_1 \parallel F_2) \geq \log\left(\frac{1 + d_{TV}}{1 - d_{TV}}\right) - \frac{2d_{TV}}{1 + d_{TV}} \geq \log\left(\frac{2 - \epsilon}{\epsilon}\right) - \frac{2 - 2\epsilon}{2 - \epsilon} \geq \log(1/\epsilon) - 1.$$

Then either

$$(\mu_1 - \mu_2)^2 \geq \log(1/\epsilon) - 1$$

or

$$(\sigma^2 - 1) - \log \sigma^2 \geq \log(1/\epsilon) - 1$$

If $\sigma \geq 1$, $\sigma^2 \geq \log(1/\epsilon) + \log \sigma^2 \geq \log(1/\epsilon)$. If $\sigma < 1$, $\log(1/\sigma^2) + \sigma^2 \geq \log(1/\epsilon)$. Then $1/\sigma^2 \geq c/\epsilon \geq \log(1/\epsilon)$ where $c < 1$ is a constant. □

Given a separation between the means, we expect that sample points originating from the same component distribution will have smaller pairwise distances than points originating from different distributions. Let X and Y be two independent samples drawn from the same F_i.

$$\begin{aligned}
\mathsf{E}\left(\|X - Y\|^2\right) &= \mathsf{E}\left(\|(X - \mu_i) - (Y - \mu_i)\|^2\right) \\
&= 2\mathsf{E}\left(\|X - \mu_i\|^2\right) - 2\mathsf{E}\left((X - \mu_i)(Y - \mu_i)\right) \\
&= 2\mathsf{E}\left(\|X - \mu_i\|^2\right) \\
&= 2\mathsf{E}\left(\sum_{j=1}^{n} |x^j - \mu_i^j|^2\right) \\
&= 2n\sigma_i^2
\end{aligned}$$

Next let X be a sample drawn from F_i and Y a sample from F_j.

$$\begin{aligned}\mathsf{E}\left(\|X-Y\|^2\right) &= \mathsf{E}\left(\|(X-\mu_i)-(Y-\mu_j)+(\mu_i-\mu_j)\|^2\right)\\ &= \mathsf{E}\left(\|X-\mu_i\|^2\right)+\mathsf{E}\left(\|Y-\mu_j\|^2\right)+\|\mu_i-\mu_j\|^2\\ &= n\sigma_i^2+n\sigma_j^2+\|\mu_i-\mu_j\|^2\end{aligned}$$

Note how this value compares to the previous one. If $\|\mu_i-\mu_j\|^2$ were large enough, points in the component with smallest variance would all be closer to each other than to any point from the other components. This suggests that we can compute pairwise distances in our sample and use them to identify the subsample from the smallest component.

We consider separation of the form

$$\|\mu_i-\mu_j\| \geq \beta \max\{\sigma_i,\sigma_j\}, \tag{2.2}$$

between every pair of means μ_i, μ_j. For β large enough, the distance between points from different components will be larger in expectation than that between points from the same component. This suggests the following classification algorithm: we compute the distances between every pair of points, and connect those points whose distance is less than some threshold. The threshold is chosen to split the graph into two (or k) cliques. Alternatively, we can compute a minimum spanning tree of the graph (with edge weights equal to distances between points), and drop the heaviest edge ($k-1$ edges) so that the graph has two (k) connected components and each corresponds to a component distribution.

Both algorithms use only the pairwise distances. In order for any algorithm of this form to work, we need to turn the above arguments about expected distance between sample points into high probability bounds. For Gaussians, we can use the following concentration bound.

Lemma 2.3. *Let X be drawn from a spherical Gaussian in $\boldsymbol{R}^n$ with mean μ and variance σ^2 along any direction. Then for any $\alpha > 1$,*

$$\mathbf{Pr}\left(\left|\|X-\mu\|^2-\sigma^2 n\right| > \alpha\sigma^2\sqrt{n}\right) \leq 2e^{-\alpha^2/8}.$$

Using this lemma with $\alpha = 4\sqrt{\ln(m/\delta)}$, to a random point X from component i, we have

$$\mathbf{Pr}(|\|X-\mu_i\|^2-n\sigma_i^2| > 4\sqrt{n\ln(m/\delta)}\sigma^2) \leq 2\frac{\delta^2}{m^2} \leq \frac{\delta}{m}$$

for $m > 2$. Thus the inequality

$$|\|X-\mu_i\|^2-n\sigma_i^2| \leq 4\sqrt{n\ln(m/\delta)}\sigma^2$$

holds for all m sample points with probability at least $1-\delta$. From this it follows that with probability at least $1-\delta$, for X, Y from the i'th and j'th Gaussians

respectively, with $i \neq j$,

$$\begin{aligned}
\|X - \mu_i\| &\leq \sqrt{\sigma_i^2 n + \alpha\sigma_i^2\sqrt{n}} \leq \sigma_i\sqrt{n} + \alpha\sigma_i \\
\|Y - \mu_j\| &\leq \sigma_j\sqrt{n} + \alpha\sigma_j \\
\|\mu_i - \mu_j\| - \|X - \mu_i\| - \|Y - \mu_j\| &\leq \|X - Y\| \leq \|X - \mu_i\| + \|Y - \mu_j\| + \|\mu_i - \mu_j\| \\
\|\mu_i - \mu_j\| - (\sigma_i + \sigma_j)(\alpha + \sqrt{n}) &\leq \|X - Y\| \leq \|\mu_i - \mu_j\| + (\sigma_i + \sigma_j)(\alpha + \sqrt{n})
\end{aligned}$$

Thus it suffices for β in the separation bound (2.2) to grow as $\Omega(\sqrt{n})$ for either of the above algorithms (clique or MST). One can be more careful and get a bound that grows only as $\Omega(n^{1/4})$ by identifying components in the order of increasing σ_i as follows.

Fact 2.4. *Suppose $X \sim N(\mu_1, \sigma_1^2 I)$ and $Y \sim N(\mu_2, \sigma_2^2 I)$ are two Gaussian random variables. Then $X + Y$ is also a Gaussian random variable and*

$$X + Y \sim N(\mu_1 + \mu_2, (\sigma_1^2 + \sigma_2^2)I).$$

For X, Y from the i'th and j'th Gaussians respectively, by the above fact, we can apply Lemma 2.3 on $X - Y$ and with probability at least $1 - 2e^{-t^2/8}$ (for any $t > 0$), we have

$$\begin{aligned}
\|X - Y\|^2 &\leq 2n\sigma_i^2 + 2t\sqrt{n}\sigma_i^2 \quad \text{if } i = j \\
\|X - Y\|^2 &> 2n\sigma_i^2 + \|\mu_i - \mu_j\|^2 - 2t\sqrt{n}\sigma_i^2 \quad \text{if } i \neq j
\end{aligned}$$

Thus it suffices to have

$$\begin{aligned}
2n\sigma_i^2 + 2t\sqrt{n}\sigma_i^2 &< 2n\sigma_i^2 + \|\mu_i - \mu_j\|^2 - 2t\sqrt{n}\sigma_i^2 \\
\|\mu_i - \mu_j\|^2 &> 2t\sqrt{n}(\sigma_i^2 + \sigma_j^2)
\end{aligned}$$

The problem with these approaches is that the separation needed grows rapidly with n, the dimension, which in general is much higher than k, the number of components. On the other hand, for classification to be achievable with high probability, the separation does not need a dependence on n. In particular, it suffices for the means to be separated by a small number of standard deviations. If such a separation holds, the projection of the mixture to the span of the means would still give a well-separate mixture and now the dimension is at most k. Of course, this is not an algorithm since the means are unknown.

One way to reduce the dimension and therefore the dependence on n is to project to a lower-dimensional subspace. A natural idea is random projection. Consider a random projection from $\boldsymbol{R}^n \to \boldsymbol{R}^\ell$ so that the image of a point u is u'. Then it can be shown that

$$\mathsf{E}\left(\|u'\|^2\right) = \frac{\ell}{n}\|u\|^2$$

In other words, the expected squared length of a vector shrinks by a factor of $\frac{\ell}{n}$. Further, the squared length is concentrated around its expectation.

$$\mathbf{Pr}(|\|u'\|^2 - \frac{\ell}{n}\|u\|^2| > \frac{\epsilon\ell}{n}\|u\|^2) \leq 2e^{-\epsilon^2\ell/4}$$

The problem with random projection is that the squared distance between the means, $\|\mu_i - \mu_j\|^2$, is also likely to shrink by the same $\frac{\ell}{n}$ factor, and therefore random projection acts only as a scaling and provides no benefit.

2.2.1 Spectral projection

Next we consider projecting to the *best-fit* subspace given by the top k singular vectors of the mixture. This is a general methodology — use principal component analysis (PCA) as a preprocessing step. In this case, it will be provably of great value.

Algorithm: Classify-Mixture

1. Compute the singular value decomposition of the sample matrix
$$A = \begin{pmatrix} x_1^T \\ \vdots \\ x_m^T \end{pmatrix}$$
2. Project the samples to the rank k subspace spanned by the top k right singular vectors.
3. Perform a distance-based classification in the k-dimensional space.

We will see that by doing this, a separation given by

$$\|\mu_i - \mu_j\| \geq c(k \log m)^{\frac{1}{4}} \max\{\sigma_i, \sigma_j\},$$

where c is an absolute constant, is sufficient for classifying m points.

The best-fit vector for a *distribution* is one that minimizes the expected squared distance of a random point to the vector. Using this definition, it is intuitive that the best fit vector for a single Gaussian is simply the vector that passes through the Gaussian's mean. We state this formally below.

Lemma 2.5. *The best-fit 1-dimensional subspace for a spherical Gaussian with mean μ is given by the vector passing through μ.*

Proof. For a randomly chosen x, we have for any unit vector v,

$$\begin{aligned} \mathsf{E}\left((x \cdot v)^2\right) &= \mathsf{E}\left(((x-\mu) \cdot v + \mu \cdot v)^2\right) \\ &= \mathsf{E}\left(((x-\mu) \cdot v)^2\right) + \mathsf{E}\left((\mu \cdot v)^2\right) + \mathsf{E}\left(2((x-\mu) \cdot v)(\mu \cdot v)\right) \\ &= \sigma^2 + (\mu \cdot v)^2 + 0 \\ &= \sigma^2 + (\mu \cdot v)^2 \end{aligned}$$

which is maximized when $v = \mu/\|\mu\|$. □

Further, due to the symmetry of the sphere, the best subspace of dimension 2 or more is *any* subspace containing the mean.

Lemma 2.6. *Any k-dimensional subspace containing μ is an optimal SVD subspace for a spherical Gaussian.*

A simple consequence of this lemma is the following theorem, which states that the best k-dimensional subspace for a mixture F involving k spherical Gaussians is the space which contains the means of the Gaussians.

Theorem 2.7. *The k-dim SVD subspace for a mixture of k Gaussians F contains the span of $\{\mu_1, \mu_2, ..., \mu_k\}$.*

Now let F be a mixture of two Gaussians. Consider what happens when we project from $\boldsymbol{R}^n$ onto the best two-dimensional subspace $\boldsymbol{R}^2$. The expected squared distance (after projection) of two points drawn from the same distribution goes from $2n\sigma_i^2$ to $4\sigma_i^2$. And, crucially, since we are projecting onto the best two-dimensional subspace which contains the two means, the expected value of $\|\mu_1 - \mu_2\|^2$ does not change!

Theorem 2.8. *Given m samples drawn from a mixture of k Gaussians with pairwise mean separation*

$$\|\mu_i - \mu_j\| \geq c(k \log m)^{\frac{1}{4}} \max\{\sigma_i, \sigma_j\}, \quad \forall i, j \in [k]$$

Classify-Mixture correctly cluster all the samples with high probability.

What property of spherical Gaussians did we use in this analysis? A spherical Gaussian projected onto the best SVD subspace is still a spherical Gaussian. In fact, this only required that the variance in every direction is equal. But many other distributions, e.g., uniform over a cube, also have this property. We address the following questions in the rest of this chapter.

1. What distributions does Theorem 2.7 extend to?
2. What about more general distributions?
3. What is the sample complexity?

2.2.2 Weakly isotropic mixtures

Next we study how our characterization of the SVD subspace can be extended.

Definition 2.9. *Random variable $X \in \mathbb{R}^n$ has a* weakly isotropic *distribution with mean μ and variance σ^2 if*

$$\mathsf{E}\left((w \cdot (X - \mu))^2\right) = \sigma^2, \quad \forall w \in \mathbb{R}^n, \ \|w\| = 1.$$

A spherical Gaussian is clearly weakly isotropic. The uniform distribution in a cube is also weakly isotropic.

Exercise 2.1. *1. Show that the uniform distribution in a cube is weakly isotropic.*

2. Show that a distribution is weakly isotropic iff its covariance matrix is a multiple of the identity.

Exercise 2.2. *The k-dimensional SVD subspace for a mixture F with component means $\mu_1, \ldots, \mu_k$ contains span$\{\mu_1, \ldots, \mu_k\}$ if each F_i is weakly isotropic.*

The statement of Exercise 2.2 does not hold for arbitrary distributions, even for $k = 1$. Consider a non-spherical Gaussian random vector $X \in \mathbb{R}^2$, whose mean is $(0, 1)$ and whose variance along the x-axis is much larger than that along the y-axis. Clearly the optimal 1-dimensional subspace for X (that maximizes the squared projection in expectation) is not the one passes through its mean μ; it is orthogonal to the mean. SVD applied after centering the mixture at the origin works for one Gaussian but breaks down for $k > 1$, even with (nonspherical) Gaussian components.

In order to demonstrate the effectiveness of this algorithm for non-Gaussian mixtures we formulate an exercise for mixtures of isotropic convex bodies.

Exercise 2.3. *Let F be a mixture of k distributions where each component is a uniform distribution over an isotropic convex body, i.e., each F_i is uniform over a convex body K_i, and satisfies*

$$\mathsf{E}_{F_i}\left((x-\mu_i)(x-\mu_i)^T\right) = I.$$

It is known that for any isotropic convex body, a random point X satisfies the following tail inequality (Lemma 2.11 later in this chapter):

$$\mathbf{Pr}(\|X - \mu_i\| > t\sqrt{n}) \leq e^{-t+1}.$$

Using this fact, derive a bound on the pairwise separation of the means of the components of F that would guarantee that spectral projection followed by distance-based classification succeeds with high probability.

2.2.3 Mixtures of general distributions

For a mixture of general distributions, the subspace that maximizes the squared projections is not the best subspace for our classification purpose any more. Consider two components that resemble "parallel pancakes", i.e., two Gaussians that are narrow and separated along one direction and spherical (and identical) in all other directions. They are separable by a hyperplane orthogonal to the line joining their means. However, the 2-dimensional subspace that maximizes the sum of squared projections (and hence minimizes the sum of squared distances) is parallel to the two pancakes. Hence after projection to this subspace, the two means collapse and we can not separate the two distributions anymore.

The next theorem provides an extension of the analysis of spherical Gaussians by showing when the SVD subspace is "close" to the subspace spanned by the component means.

Theorem 2.10. *Let F be a mixture of arbitrary distributions $F_1, \ldots, F_k$. Let w_i be the mixing weight of F_i, μ_i be its mean and $\sigma^2_{i,W}$ be the maximum variance of F_i along directions in W, the k-dimensional SVD-subspace of F. Then*

$$\sum_{i=1}^{k} w_i d(\mu_i, W)^2 \leq k \sum_{i=1}^{k} w_i \sigma^2_{i,W}$$

where $d(.,.)$ is the orthogonal distance.

Theorem 2.10 says that for a mixture of general distributions, the means do not move too much after projection to the SVD subspace. Note that the theorem does not solve the case of parallel pancakes, as it requires that the pancakes be separated by a factor proportional to their "radius" rather than their "thickness".

Proof. Let M be the span of $\mu_1, \mu_2, \ldots, \mu_k$. For $x \in \boldsymbol{R}^n$, we write $\pi_M(x)$ for the projection of x to the subspace M and $\pi_W(x)$ for the projection of x to W.

We first lower bound the expected squared length of the projection to the mean subpspace M.

$$\begin{aligned}
\mathsf{E}\left(\|\pi_M(x)\|^2\right) &= \sum_{i=1}^{k} w_i \mathsf{E}_{F_i}\left(\|\pi_M(x)\|^2\right) \\
&= \sum_{i=1}^{k} w_i \left(\mathsf{E}_{F_i}\left(\|\pi_M(x) - \mu_i\|^2\right) + \|\mu_i\|^2\right) \\
&\geq \sum_{i=1}^{k} w_i \|\mu_i\|^2 \\
&= \sum_{i=1}^{k} w_i \|\pi_W(\mu_i)\|^2 + \sum_{i=1}^{k} w_i d(\mu_i, W)^2.
\end{aligned}$$

We next upper bound the expected squared length of the projection to the SVD subspace W. Let $\vec{e}_1, ..., \vec{e}_k$ be an orthonormal basis for W.

$$\begin{aligned}
\mathsf{E}\left(\|\pi_W(x)\|^2\right) &= \sum_{i=1}^{k} w_i \left(\mathsf{E}_{F_i}\left(\|\pi_W(x - \mu_i)\|^2\right) + \|\pi_W(\mu_i)\|^2\right) \\
&\leq \sum_{i=1}^{k} w_i \sum_{j=1}^{k} \mathsf{E}_{F_i}\left((\pi_W(x - \mu_i) \cdot \vec{e}_j)^2\right) + \sum_{i=1}^{k} w_i \|\pi_W(\mu_i)\|^2 \\
&\leq k \sum_{i=1}^{k} w_i \sigma^2_{i,W} + \sum_{i=1}^{k} w_i \|\pi_W(\mu_i)\|^2.
\end{aligned}$$

The SVD subspace maximizes the sum of squared projections among all subspaces of rank at most k (Theorem 1.3). Therefore,

$$\mathsf{E}\left(\|\pi_M(x)\|^2\right) \leq \mathsf{E}\left(\|\pi_W(x)\|^2\right)$$

and the theorem follows from the previous two inequalities. □

The next exercise gives a refinement of this theorem.

Exercise 2.4. *Let S be a matrix whose rows are a sample of m points from a mixture of k distributions with m_i points from the i'th distribution. Let $\bar{\mu}_i$ be the mean of the subsample from the i'th distribution and $\bar{\sigma}_i^2$ be its largest directional variance. Let W be the k-dimensional SVD subspace of S.*

1. *Prove that*
$$\|\bar{\mu}_i - \pi_W(\bar{\mu}_i)\| \leq \frac{\|S - \pi_W(S)\|}{\sqrt{m_i}}$$
where the norm on the RHS is the 2-norm (largest singular value).

2. *Let $\bar{S}$ denote the matrix where each row of S is replaced by the corresponding $\bar{\mu}_i$. Show that (again with 2-norm),*
$$\|S - \bar{S}\|^2 \leq \sum_{i=1}^{k} m_i \bar{\sigma}_i^2.$$

3. *From the above, derive that for each component,*
$$\|\bar{\mu}_i - \pi_W(\bar{\mu}_i)\|^2 \leq \frac{\sum_{j=1}^{k} w_j \bar{\sigma}_j^2}{w_i}$$
where $w_i = m_i/m$.

2.2.4 Spectral projection with samples

So far we have shown that the SVD subspace of a mixture can be quite useful for classification. In reality, we only have samples from the mixture. This section is devoted to establishing bounds on sample complexity to achieve similar guarantees as we would for the full mixture. The main tool will be distance concentration of samples. In general, we are interested in inequalities such as the following for a random point X from a component F_i of the mixture. Let $R^2 = \mathsf{E}\left(\|X - \mu_i\|^2\right)$.

$$\mathbf{Pr}\left(\|X - \mu_i\| > tR\right) \leq e^{-ct}.$$

This is useful for two reasons:

1. To ensure that the SVD subspace the sample matrix is not far from the SVD subspace for the full mixture. Since our analysis shows that the SVD subspace is near the subspace spanned by the means and the distance, all we need to show is that the sample means and sample variances converge to the component means and covariances.

2. To be able to apply simple clustering algorithms such as forming cliques or connected components, we need distances between points of the same component to be not much higher than their expectations.

An interesting general class of distributions with such concentration properties are those whose probability density functions are *logconcave*. A function f is logconcave if $\forall x, y,\ \forall \lambda \in [0,1]$,

$$f(\lambda x + (1-\lambda)y) \geq f(x)^{\lambda} f(y)^{1-\lambda}$$

or equivalently,

$$\log f(\lambda x + (1-\lambda)y) \geq \lambda \log f(x) + (1-\lambda)\log f(y).$$

Many well-known distributions are log-concave. In fact, any distribution with a density function $f(x) = e^{g(x)}$ for some concave function $g(x)$, e.g. $e^{-c\|x\|}$ or $e^{c(x\cdot v)}$ is logconcave. Also, the uniform distribution in a convex body is logconcave. The following concentration inequality [LV07] holds for any logconcave density.

Lemma 2.11. *Let X be a random point from a logconcave density in $\boldsymbol{R}^n$ with $\mu = \mathsf{E}(X)$ and $R^2 = \mathsf{E}(\|X-\mu\|^2)$. Then,*

$$\mathbf{Pr}(\|X-\mu\| \geq tR) \leq e^{-t+1}.$$

Putting this all together, we conclude that Algorithm *Classify-Mixture*, which projects samples to the SVD subspace and then clusters, works well for mixtures of well-separated distributions with logconcave densities, where the separation required between every pair of means is proportional to the largest standard deviation.

Theorem 2.12. *Algorithm* Classify-Mixture *correctly classifies a sample of m points from a mixture of k arbitrary logconcave densities $F_1, \ldots, F_k$, with probability at least $1-\delta$, provided for each pair i, j we have*

$$\|\mu_i - \mu_j\| \geq Ck^c \log(m/\delta) \max\{\sigma_i, \sigma_j\},$$

μ_i is the mean of component F_i, σ_i^2 is its largest variance and c, C are fixed constants.

This is essentially the best possible guarantee for the algorithm. However, it is a bit unsatisfactory since an affine transformation, which does not affect probabilistic separation, could easily turn a well-separated mixture into one that is not well-separated.

2.3 Learning mixtures of spherical distributions

So far our efforts have been to partition the observed sample points. The other interesting problem proposed in the introduction of this chapter was to identify

the values μ_i, σ_i and w_i. In this section, we will see that this is possible in polynomial time provided the means μ_i are linearly independent. We let Y denote a sample from the mixture F. Thus,

$$\begin{aligned} \mathsf{E}(Y) &= \sum_i w_i \mathsf{E}_{F_i}(X) = \sum_i w_i \mu_i \\ \mathsf{E}(Y \otimes Y)_{jk} &= \mathsf{E}(Y_j Y_k) \end{aligned}$$

Before we go on, let us clarify some notation. The operator $\otimes$ is the tensor product; for vectors u, v, we have $u \otimes v = uv^T$. Note how u and v are vectors but uv^T is a matrix. For a tensor product between a matrix and a vector, say $A \otimes u$, the result is a tensor with three dimensions. In general, the resulting dimensionality is the sum of the argument dimensions.

Next we derive an expression for the second moment tensor. For $X \sim \mathcal{N}(\mu, \sigma^2 I)$,

$$\begin{aligned} \mathsf{E}(X \otimes X) &= \mathsf{E}((X - \mu + \mu) \otimes (X - \mu + \mu)) \\ &= \mathsf{E}((X - \mu) \otimes (X - \mu)) + \mu \otimes \mu \\ &= \sigma^2 I + \mu \otimes \mu. \end{aligned}$$

Therefore,

$$\begin{aligned} \mathsf{E}(Y \otimes Y) &= \sum_i w_i \mathsf{E}_i(X \otimes X) \\ &= (\sum_i w_i \sigma_i^2) I + \sum_i w_i (\mu_i \otimes \mu_i) \end{aligned}$$

where $\mathsf{E}_i(.) = \mathsf{E}_{F_i}(.)$.

Let us now see what happens if we take the inner product of Y and some vector v.

$$\mathsf{E}((Y \cdot v)^2) = v^T \mathsf{E}(Y \otimes Y) v = \sum_i w_i \sigma_i^2 + \sum_i w_i (\mu_i^T v)^2.$$

One observation is that if v were orthogonal to $\text{span}\{\mu_1 \dots \mu_k\}$, then we would have:

$$\mathsf{E}((Y \cdot v)^2) = \sum_i w_i \sigma_i^2$$

Therefore we can compute

$$M = \mathsf{E}(Y \otimes Y) - \mathsf{E}((Y \cdot v)^2) I = \sum_{i=1}^{k} w_i \mu_i \otimes \mu_i.$$

We do not know the μ_i's, so finding a v orthogonal to them is not straightforward. However, if we compute the SVD of the $m \times n$ matrix containing our

m samples, the top k singular vectors would be the best fit k-dimensional subspace (see theorem 1.4), and assuming the means are linearly independent, this is exactly span$\{\mu_1 \dots \mu_k\}$.

Exercise 2.5. *Show that for any $j > k$, the j'th singular value σ_j is equal to $\sum_i w_i \sigma_i^2$ and the corresponding right singular vector is orthogonal to span$\{\mu_1, \dots, \mu_k\}$.*

Exercise 2.6. *Show that it is possible for two mixtures with distinct sets of means to have exactly the same second moment tensor.*

From the exercises above, it should now be clear that the calculations for the second moments are not enought to retrieve the distribution parameters. It might be worth experimenting with the third moment, so let us calculate $\mathsf{E}\,(Y \otimes Y \otimes Y)$.

For $X \sim \mathcal{N}(\mu, \sigma^2 I)$,

$$\begin{aligned}
& \mathsf{E}\,(X \otimes X \otimes X) \\
= \;& \mathsf{E}\,((X-\mu+\mu) \otimes (X-\mu+\mu) \otimes (X-\mu+\mu)) \\
= \;& \mathsf{E}\,((X-\mu) \otimes (X-\mu) \otimes (X-\mu)) \\
+ \;& \mathsf{E}\,(\mu \otimes (X-\mu) \otimes (X-\mu) + (X-\mu) \otimes \mu \otimes (X-\mu) + (X-\mu) \otimes (X-\mu) \otimes \mu) \\
+ \;& \mathsf{E}\,((X-\mu) \otimes \mu \otimes \mu + \mu \otimes (X-\mu) \otimes \mu + \mu \otimes \mu \otimes (X-\mu)) \\
+ \;& \mathsf{E}\,(\mu \otimes \mu \otimes \mu) \\
& \quad \text{(here we have used the fact that the odd powers of } (X-\mu) \text{ have mean zero)} \\
= \;& \mathsf{E}\,(\mu \otimes (X-\mu) \otimes (X-\mu)) + \mathsf{E}\,((X-\mu) \otimes \mu \otimes (X-\mu)) \\
+ \;& \mathsf{E}\,((X-\mu) \otimes (X-\mu) \otimes \mu) + \mathsf{E}\,(\mu \otimes \mu \otimes \mu) \\
= \;& \mu \otimes \sigma^2 I + \sigma^2 \sum_j^n e_j \otimes \mu \otimes e_j + \sigma^2 I \otimes \mu + \mu \otimes \mu \otimes \mu \\
= \;& \sigma^2 \sum_j^n \mu \otimes e_j \otimes e_j + e_j \otimes \mu \otimes e_j + e_j \otimes e_j \otimes \mu + \mu \otimes \mu \otimes \mu.
\end{aligned}$$

Then the third moment for Y can be expressed as

$$\begin{aligned}
\mathsf{E}\,(Y \otimes Y \otimes Y) \;=\; & \sum_i^k w_i \sigma_i^2 \left(\sum_j^n \mu_i \otimes e_j \otimes e_j + e_j \otimes \mu_i \otimes e_j + e_j \otimes e_j \otimes \mu_i \right) \\
+ \; & \sum_i^k w_i \mu_i \otimes \mu_i \otimes \mu_i
\end{aligned}$$

However, we must not forget that we haven't really made any progress unless our subexpressions are estimable using sample points. When we were doing calculations for the second moment, we could in the end estimate $\sum_i w_i \mu_i \otimes \mu_i$ from $\mathsf{E}\,(Y \otimes Y)$ and $\sum_i w_i \sigma_i^2$ using the SVD, see Exercise 2.5. Similarly, we are now going to show that we'll be able to estimate $\sum_i^k w_i \sigma^2 \sum_j^n \mu_i \otimes e_j \otimes e_j + e_j \otimes$

$\mu_i \otimes e_j + e_j \otimes e_j \otimes \mu_i$ and hence also $\sum_i w_i \mu_i \otimes \mu_i \otimes \mu_i$. We will use the same idea of having a vector v orthogonal to the span of the means.

First, for any unit vector v and $X \sim \mathcal{N}(\mu, \sigma^2 I)$,

$$\begin{aligned}
\mathsf{E}\left(X((X-\mu)\cdot v)^2\right) &= \mathsf{E}\left((X-\mu+\mu)((X-\mu)\cdot v)^2\right) \\
&= \mathsf{E}\left((X-\mu)((X-\mu)\cdot v)^2\right) + \mathsf{E}\left(\mu((X-\mu)\cdot v)^2\right) \\
&= 0 + \mathsf{E}\left(\mu((X-\mu)\cdot v)^2\right) \\
&= \mathsf{E}\left(((X-\mu)\cdot v)^2\right)\mu \\
&= \sigma^2 \mu
\end{aligned}$$

Before going to the Y case, lets introduce a convenient notation of treating tensors as functions, say for a third-order tensor T, we define these three functions on it

$$T(x,y,z) = Txyz = \sum_{jkl} T_{jkl} x_j y_k z_l$$

In particular we note that for vectors a and b, $(a \otimes a \otimes a)(b,b)) = \left(a(a\cdot b)^2\right)$. With this in mind we continue to explore the third moment.

$$\begin{aligned}
\mathsf{E}\left(Y((Y-\mu_Y)\cdot v)^2\right) &= \mathsf{E}\left(Y \otimes (Y-\mu_Y) \otimes (Y-\mu_Y)\right)(v,v) \\
&= \mathsf{E}\left(Y \otimes Y \otimes Y\right)(v,v) + \mathsf{E}\left(Y \otimes Y \otimes -\mu_Y\right)(v,v) \\
&+ \mathsf{E}\left(Y \otimes -\mu_Y \otimes (Y-\mu_Y)\right)(v,v)
\end{aligned}$$

Now assume that v is perpendicular to the means. Therefore μ_Y is also perpendicular to v because it must be in $\text{span}\{\mu_1 \dots \mu_k\}$.

$$\begin{aligned}
& \mathsf{E}\left(Y((Y-\mu_Y)\cdot v)^2\right) \\
= \; & \mathsf{E}\left(Y \otimes Y \otimes Y\right)(v,v) + \mathsf{E}\left(Y \otimes Y \otimes -\mu_Y\right)(v,v) \\
+ \; & \mathsf{E}\left(Y \otimes -\mu_Y \otimes (Y-\mu_Y)\right)(v,v) \\
= \; & \mathsf{E}\left(Y \otimes Y \otimes Y\right)(v,v) \\
= \; & \sum_i^k w_i \sigma_i^2 \left(\sum_j^n \mu_i \otimes e_j \otimes e_j + e_j \otimes \mu_i \otimes e_j + e_j \otimes e_j \otimes \mu_i \right)(v,v) \\
+ \; & \sum_i^k w_i \mu_i \otimes \mu_i \otimes \mu_i (v,v) \\
= \; & \sum_i^k w_i \sigma_i^2 \left(\sum_j^n \mu_i \otimes v \otimes v \right) \\
= \; & \sum_i^k w_i \sigma_i^2 \mu_i
\end{aligned}$$

Now, let's form the expression $u = \mathsf{E}\left(Y((Y-\mu_Y)\cdot v)^2\right)$ where μ_Y is the mean of Y. Note that u is a estimable vector and also parametrized over v. And since u is estimable, so is $\sum_j (u \otimes e_j \otimes e_j + e_j \otimes u \otimes e_j + e_j \otimes e_j \otimes u)$.

$$\begin{aligned} T &= \mathsf{E}\left(Y \otimes Y \otimes Y\right) - \sum_j (u \otimes e_j \otimes e_j + e_j \otimes u \otimes e_j + e_j \otimes e_j \otimes u) \\ &= \sum_i w_i(\mu_i \otimes \mu_i \otimes \mu_i). \end{aligned}$$

So far we have seen how to compute $M = \sum_i w_i \mu_i \otimes \mu_i$ and T above from samples. We are now ready to state the algorithm. For a set of samples S and any function on $\boldsymbol{R}^n$, let $E_S(f(x))$ denote the average of f over points in S.

Algorithm: Learning the parameters

1. Compute $M = E_S(Y \otimes Y)$ and its top k eigenvectors $v_1, \ldots v_k$. Let $\bar{\sigma} = \sigma_{k+1}(M)$.
2. Project the data to the span of $v_1, \ldots, v_k$. Decompose $(M - \bar{\sigma} I) = WW^T$, using the SVD, and compute $\tilde{S} = W^{-1}S$.
3. Find a vector $\bar{v}$ orthogonal to span$\{W^{-1}v_1, \ldots W^{-1}v_k\}$ and compute
$$\bar{u} = E_{\tilde{S}}(Y((Y-\mu_Y)\bar{v})^2)$$
and
$$T = E_{\tilde{S}}(Y \otimes Y \otimes Y) - \sum_j (\bar{u} \otimes e_j \otimes e_j + e_j \otimes \bar{u} \otimes e_j + e_j \otimes e_j \otimes \bar{u}).$$
4. Iteratively apply the tensor power method on T. That is repeteadly apply
$$x := \frac{T(.,x,x)}{\|T(.,x,x)\|}$$
until convergence. Then set $\tilde{\mu}_1 = T(x,x,x)x$ and $w_1 = 1/|\tilde{\mu}_1|^2$ and repeat with
$$T := T - w_i \tilde{\mu}_i \otimes \tilde{\mu}_i \otimes \tilde{\mu}_i$$
to recover $\tilde{\mu}_2 \ldots \tilde{\mu}_k$. To recover the variances, compute $\sigma_i^2 = \bar{u} \cdot \tilde{\mu}_i$.

The algorithm's performance is analyzed in the following theorem.

Theorem 2.13. *Given M and T as above, if all the means $\mu_1 \ldots \mu_k$ are linearly independent, we can estimate all parameters of each distribution in polynomial time.*

Make M isotropic by decomposing it into $M = WW^T$. Let $\tilde{\mu}_i = W^{-1}\mu_i$. From this definition we have

$$\begin{aligned}\sum_i w_i(\tilde{\mu}_i \otimes \tilde{\mu}_i) &= \sum_i w_i(W^{-1}\mu_i)(W^{-1}\mu_i)^T \\ &= W^{-1}(\sum_i w_i\mu_i\mu_i^T){W^{-1}}^T \\ &= W^{-1}B_2{W^{-1}}^T \\ &= I\end{aligned}$$

Exercise 2.7. *Show that the $\sqrt{w_i}\tilde{\mu}_i$ are orthonormal.*

Now for the third-order tensor

$$T = \sum_i w_i(\tilde{\mu}_i \otimes \tilde{\mu}_i \otimes \tilde{\mu}_i)$$

we have,

$$\begin{aligned}T(x,x,x) &= \sum_{jkl} T_{jkl}x_jx_kx_l \\ &= \sum_{jkl}(\sum_i w_i(\tilde{\mu}_i \otimes \tilde{\mu}_i \otimes \tilde{\mu}_i))x_jx_kx_l \\ &= \sum_i w_i(\tilde{\mu}_i \cdot x)^3\end{aligned}$$

Now we apply Theorem 7.1 to conclude that when started at a random x, with high probability, the tensor power method converges to one of the $\tilde{\mu}_i$.

2.4 An affine-invariant algorithm

We now return to the general mixtures problem, seeking a better condition on separation than that we derived using spectral projection. The algorithm described here is an application of isotropic PCA, an algorithm discussed in Chapter **??**. Unlike the methods we have seen so far, the algorithm is affine-invariant. For $k = 2$ components it has nearly the best possible guarantees for clustering Gaussian mixtures. For $k > 2$, it requires that there be a $(k-1)$-dimensional subspace where the *overlap* of the components is small in every direction. This condition can be stated in terms of the Fisher discriminant, a quantity commonly used in the field of Pattern Recognition with labeled data. The affine invariance makes it possible to unravel a much larger set of Gaussian mixtures than had been possible previously. Here we only describe the case of two components in detail, which contains the key ideas.

The first step of the algorithm is to place the mixture in isotropic position via an affine transformation. This has the effect of making the $(k-1)$-dimensional

Fisher subspace, i.e., the one that minimizes the Fisher discriminant (the fraction of the variance of the mixture taken up the intra-component term; see Section 2.4.2 for a formal definition), the same as the subspace spanned by the means of the components (they only coincide in general in isotropic position), for *any* mixture. The rest of the algorithm identifies directions close to this subspace and uses them to cluster, without access to labels. Intuitively this is hard since after isotropy, standard PCA/SVD reveals no additional information. Before presenting the ideas and guarantees in more detail, we describe relevant related work.

As before, we assume we are given a lower bound w on the minimum mixing weight and k, the number of components. With high probability, Algorithm UNRAVEL returns a hyperplane so that each halfspace encloses almost all of the probability mass of a single component and almost none of the other component.

The algorithm has three major components: an initial affine transformation, a reweighting step, and identification of a direction close to the Fisher direction. The key insight is that the reweighting technique will either cause the mean of the mixture to shift in the intermean subspace, or cause the top principal component of the second moment matrix to approximate the intermean direction. In either case, we obtain a direction along which we can partition the components.

We first find an affine transformation W which when applied to $\mathcal{F}$ results in an isotropic distribution. That is, we move the mean to the origin and apply a linear transformation to make the covariance matrix the identity. We apply this transformation to a new set of m_1 points $\{x_i\}$ from $\mathcal{F}$ and then reweight according to a spherically symmetric Gaussian $\exp(-\|x\|^2/\alpha)$ for $\alpha = \Theta(n/w)$. We then compute the mean $\hat{u}$ and second moment matrix $\hat{M}$ of the resulting set. After the reweighting, the algorithm chooses either the new mean or the direction of maximum second moment and projects the data onto this direction h.

Algorithm Unravel
Input: Scalar $w > 0$.
Initialization: $P = \mathbb{R}^n$.

1. (Rescale) Use samples to compute an affine transformation W that makes the distribution nearly isotropic (mean zero, identity covariance matrix).
2. (Reweight) For each of m_1 samples, compute a weight $e^{-\|x\|^2/\alpha}$.
3. (Find Separating Direction) Find the mean of the reweighted data $\hat{\mu}$. If $\|\hat{\mu}\| > \sqrt{w}/(32\alpha)$ (where $\alpha > n/w$), let $h = \hat{\mu}$. Otherwise, find the covariance matrix $\hat{M}$ of the reweighted points and let h be its top principal component.
4. (Classify) Project m_2 sample points to h and classify the projection based on distances.

2.4.1 Parallel Pancakes

We now discuss the case of parallel pancakes in detail. Suppose $\mathcal{F}$ is a mixture of two spherical Gaussians that are well-separated, i.e. the intermean distance is large compared to the standard deviation along any direction. We consider two cases, one where the mixing weights are equal and another where they are imbalanced.

After isotropy is enforced, each component will become thin in the intermean direction, giving the density the appearance of two parallel pancakes. When the mixing weights are equal, the means of the components will be equally spaced at a distance of $1 - \phi$ on opposite sides of the origin. For imbalanced weights, the origin will still lie on the intermean direction but will be much closer to the heavier component, while the lighter component will be much further away. In both cases, this transformation makes the variance of the mixture 1 in every direction, so the principal components give us no insight into the inter-mean direction.

Consider next the effect of the reweighting on the mean of the mixture. For the case of equal mixing weights, symmetry assures that the mean does not shift at all. For imbalanced weights, however, the heavier component, which lies closer to the origin will become heavier still. Thus, the reweighted mean shifts toward the mean of the heavier component, allowing us to detect the intermean direction.

Finally, consider the effect of reweighting on the second moments of the mixture with equal mixing weights. Because points closer to the origin are weighted more, the second moment in every direction is reduced. However, in the intermean direction, where part of the moment is due to the displacement of the component means from the origin, it shrinks less. Thus, the direction of

maximum second moment is the intermean direction.

2.4.2 Analysis

The algorithm has the following guarantee for a two-Gaussian mixture.

Theorem 2.14. *Let w_1, μ_1, Σ_1 and w_2, μ_2, Σ_2 define a mixture of two Gaussians and $w = \min w_1, w_2$. There is an absolute constant C such that, if there exists a direction v such that*

$$|\pi_v(\mu_1 - \mu_2)| \geq C \left(\sqrt{v^T \Sigma_1 v} + \sqrt{v^T \Sigma_2 v} \right) w^{-2} \log^{1/2} \left(\frac{1}{w\delta} + \frac{1}{\eta} \right),$$

then with probability $1 - \delta$ algorithm Unravel *returns two complementary halfspaces that have error at most η using time and a number of samples that is polynomial in $n, w^{-1}, \log(1/\delta)$.*

So the separation required between the means is comparable to the standard deviation in *some direction*. This separation condition of Theorem 2.14 is affine-invariant and much weaker than conditions of the form $\|\mu_1 - \mu_2\| \gtrsim \max\{\sigma_{1,\max}, \sigma_{2,\max}\}$ that came up earlier in the chapter. We note that the separating direction need not be the intermean direction.

It will be insightful to state this result in terms of the Fisher discriminant, a standard notion from Pattern Recognition [DHS01, Fuk90] that is used with labeled data. In words, the Fisher discriminant along direction p is

$$J(p) = \frac{\text{the intra-component variance in direction } p}{\text{the total variance in direction } p}$$

Mathematically, this is expressed as

$$J(p) = \frac{E\left[\|\pi_p(x - \mu_{\ell(x)})\|^2\right]}{E\left[\|\pi_p(x)\|^2\right]} = \frac{p^T(w_1\Sigma_1 + w_2\Sigma_2)p}{p^T(w_1(\Sigma_1 + \mu_1\mu_1^T) + w_2(\Sigma_2 + \mu_2\mu_2^T))p}$$

for x distributed according to a mixture distribution with means μ_i and covariance matrices Σ_i. We use $\ell(x)$ to indicate the component from which x was drawn.

Theorem 2.15. *There is an absolute constant C for which the following holds. Suppose that $\mathcal{F}$ is a mixture of two Gaussians such that there exists a direction p for which*

$$J(p) \leq Cw^3 \log^{-1} \left(\frac{1}{\delta w} + \frac{1}{\eta} \right).$$

With probability $1 - \delta$, algorithm Unravel *returns a halfspace with error at most η using time and sample complexity polynomial in $n, w^{-1}, \log(1/\delta)$.*

In words, the algorithm successfully unravels arbitrary Gaussians provided there exists a line along which the expected squared distance of a point to its component mean is smaller than the expected squared distance to the overall

mean by roughly a $1/w^3$ factor. There is no dependence on the largest variances of the individual components, and the dependence on the ambient dimension is logarithmic. Thus the addition of extra dimensions, even with large variance, has little impact on the success of the algorithm. The algorithm and its analysis in terms of the Fisher discriminant have been generalized to $k > 2$ [BV08].

2.5 Discussion

Mixture models are a classical topic in statistics. Traditional methods such as EM or other local search heuristics can get stuck in local optima or take a long time to converge. Starting with Dasgupta's paper [Das99] in 1999, there has been much progress on efficient algorithms with rigorous guarantees [AK05, DS00], with Arora and Kannan [AK05] addressing the case of general Gaussians using distance concentration methods. PCA was analyzed in this context by Vempala and Wang [VW04] giving nearly optimal guarantees for mixtures of spherical Gaussians (and weakly isotropic distributions). This was extended to general Gaussians and logconcave densities [KSV08, AM05] (Exercise 2.4 is based on [AM05]), although the bounds obtained were far from optimal in that the separation required grows with the largest variance of the components or with the dimension of the underlying space. In 2008, Brubaker and Vempala [BV08] presented an affine-invariant algorithm that only needs hyperplane separability for two Gaussians and a generalization of this condition for $k > 2$; in particular, it suffices for each component to be separable from the rest of the mixture by a hyperplane.

A related line of work considers learning symmetric product distributions, where the coordinates are independent. Feldman et al [FSO06] have shown that mixtures of axis-aligned Gaussians can be approximated without any separation assumption at all in time exponential in k. Chaudhuri and Rao [CR08a] have given a polynomial-time algorithm for clustering mixtures of product distributions (axis-aligned Gaussians) under mild separation conditions. A. Dasgupta et al [DHKS05] and later Chaudhuri and Rao [CR08b] gave algorithms for clustering mixtures of heavy-tailed distributions.

For learning all parameters of a mixture of two Gaussians, Kalai, Moitra and Valiant [KMV10] gave a polynomial-time algorithm with no separation requirement. This was lated extended to a mixture of k Gaussians with sample and time complexity $n^{f(k)}$ by Moitra and Valiant [MV10]. For arbitrary k-Gaussian mixtures, they also show a lower bound of $2^{\Omega(k)}$ on the sample complexity.

In 2012, Hsu and Kakade [HK13] found the method described here for learning parameters of a mixture of spherical Gaussians assuming only that their means are linearly independent. It is an open problem to extend their approach to a mixture of general Gaussians under suitable nondegeneracy assumptions (perhaps the same).

A more general question is "agnostic" learning of Gaussians, where we are given samples from an arbitrary distribution and would like to find the best-fit mixture of k Gaussians. This problem naturally accounts for noise and

appears to be much more realistic. Brubaker [Bru09] gave an algorithm that makes progress towards this goal, by allowing a mixture to be corrupted by an ϵ fraction of noisy points with $\epsilon < w_{\min}$, and with nearly the same separation requirements as in Section 2.2.3.

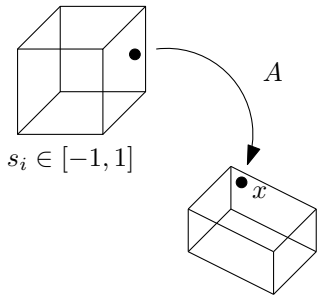
Chapter 3

Independent Component Analysis

Suppose that $s \in \mathbf{R}^n$, $s = (s_1, s_2, \ldots, s_n)$, is a vector of independent signals (or components) that we cannot directly measure. However, we are able to gather a set of samples, $x = (x_1, x_2, \ldots, x_k)$, where

$$x = As,$$

i.e., the x_i's are a sampling ($k \leq n$) of the linearly transformed signals. Given $x = As$, where $x \in \mathbf{R}^n$, A is full rank, and $s_1, s_2, \ldots, s_n$ are independent. We want to identify or estimate A from the samples x. The goal of Independent Component Analysis (ICA) is to recover the unknown linear transformation A.



A simple example application of ICA is shown in Fig. 3. Suppose that the signals s_i are points in some space (e.g., a three-dimensional cube) and we want to find the linear transformation A that produces the points x in some other transformed space. We can only recover A by sampling points in the transformed space, since we cannot directly measure points in the original space.

Another example of an application of ICA is the *cocktail party* problem, where several microphones are placed throughout a room in which a party is

held. Each microphone is able to record the conversations nearby and the problem is to recover the words that are spoken by each person in the room from the overlapping conversations that are recorded.

3.1 Recovery with fourth moment assumptions

We want to know whether it is possible to recover A. The following algorithm will recover A under a condition on the first four moments of the components of s.

Algorithm: ICA

1. Make the samples isotropic.
2. Form the 4'th order tensor M with entries

   $$M_{i,j,k,l} = \mathsf{E}(x_i x_j x_k x_l) - \mathsf{E}(x_i x_j)\mathsf{E}(x_k x_l) - \mathsf{E}(x_i x_k)\mathsf{E}(x_j x_l) - \mathsf{E}(x_i x_l)\mathsf{E}(x_j x_k).$$

3. Apply Tensor Power Iteration to M and return the vectors obtained.

Tensor Power Iteration is an extension of matrix power iteration. For a symmetric tensor T, the iteration starts with a random unit vector y^0 and applies

$$y^{i+1} = \frac{T(y^i, y^i, y^i, .)}{\|T(y^i, y^i, y^i, .)\|}$$

where $T(u, v, w, .)_i = \sum_{jkl} T_{ijkl} u_j v_k w_l$.

Theorem 3.1. *Assume that*

1. $\mathsf{E}(ss^T) = I$.
2. $\mathsf{E}(s_i^4) = 3$ *for at most one component* i.
3. A *is* $n \times n$ *and full rank.*

Then with high probability, Algorithm ICA will recover the columns of A *to any desired accuracy* ϵ *using time and samples polynomial in* $n, 1/\epsilon, 1/\sigma_{min}$ *where* σ_{min} *is the smallest singular value of* A.

Proof. The first moment is $\mathsf{E}(x) = A\mathsf{E}(s) = 0$. The second moment is,

$$\begin{aligned} \mathsf{E}(xx^T) &= \mathsf{E}(As(As)^T) \\ &= A\mathsf{E}(ss^T)A^T \\ &= AA^T = \sum_{i=1}^{n} A_{(i)} \otimes A_{(i)} \end{aligned}$$

The third moment, $\mathsf{E}\,(x \otimes x \otimes x)$ could be zero, so we examine the fourth moment.

$$\begin{aligned}
\mathsf{E}\,(x \otimes x \otimes x \otimes x)_{i,j,k,l} &= \mathsf{E}\,(x_i x_j x_k x_l) \\
&= \mathsf{E}\,((As)_i (As)_j (As)_k (As)_l) \\
&= \mathsf{E}\,\left((A_{(i)} \cdot s)(A_j \cdot s)(A_k \cdot s)(A_{(l)} \cdot s)\right) \\
&= \mathsf{E}\,\left(\sum_{i'} A_{ii'} s_{i'} \sum_{j'} A_{jj'} s_{j'} \sum_{k'} A_{kk'} s_{k'} \sum_{l'} A_{ll'} s_{l'}\right) \\
&= \sum_{i',j',k',l'} A_{ii'} A_{jj'} A_{kk'} A_{ll'} \mathsf{E}\,(s_{i'} s_{j'} s_{k'} s_{l'})
\end{aligned}$$

Based on the assumptions about the signal,

$$\mathsf{E}\,(s_{i'} \dots s_{l'}) = \begin{cases} \mathsf{E}\,(s_{i'}^4) & \text{if } s_{i'} = \dots = s_{l'} \\ 1 & \text{if } s_{i'} = s_{j'} \neq s_{k'} = s_{l'} \\ 0 & \text{otherwise,} \end{cases}$$

which we can plug back into the previous equation.

$$\begin{aligned}
\mathsf{E}\,(x \otimes x \otimes x \otimes x)_{i,j,k,l} &= \sum_{i',j'} \left(A^{(i')} \otimes A^{(i')} \otimes A^{(j')} \otimes A^{(j')}\right) \mathsf{E}\,(s_{i'}^2 s_{j'}^2) \\
&+ \sum_{i',j'} \left(A^{(i')} \otimes A^{(j')} \otimes A^{(j')} \otimes A^{(i')}\right) \mathsf{E}\,(s_{i'}^2 s_{j'}^2) \\
&+ \sum_{i',j'} \left(A^{(i')} \otimes A^{(j')} \otimes A^{(i')} \otimes A^{(j')}\right) \mathsf{E}\,(s_{i'}^2 s_{j'}^2).
\end{aligned}$$

Now, if we apply

$$\mathsf{E}\,(s_{i'}^2 s_{j'}^2) = \begin{cases} 1 & \text{if } i' \neq j' \\ \mathsf{E}\,(s_{i'}^4) & \text{otherwise.} \end{cases}$$

We define the tensor

$$(M_1)_{i,j,k,l} = \mathsf{E}\,(x_i x_j)\mathsf{E}\,(x_k x_l) + \mathsf{E}\,(x_i x_k)\mathsf{E}\,(x_j x_l) + \mathsf{E}\,(x_i x_l)\mathsf{E}\,(x_j x_k),$$

Then,

$$M = \mathsf{E}\,(\otimes^4 x) - M_1 = \sum_{i'} (\mathsf{E}\,(s_i'^4) - 3) A^{(i')} \otimes A^{(i')} \otimes A^{(i')} \otimes A^{(i')}$$

is a linear combination of outer products of orthogonal tensors. By our assumptions, the coefficients are nonzero except for at most one term. The tensor itself can be estimated to arbitrary accuracy with polynomially many samples.Tensor Power Iteration then recovers the decomposition to any desired accuracy. □

Exercise 3.1. *Show that A is not uniquely identifiable if the distributions of two or more signals s_i are Gaussian. Show that if only one component is Gaussian, then A can still be recovered.*

We have shown that we can use ICA to uniquely recover A if the distribution of no more than one of the signals is Gaussian. One problem is that the fourth moment tensor has size n^4. However, we can avoid constructing the tensor explicitly.

For any vector u,

$$M(u,u)_{i,j} = \sum_{k,l} M_{i,j,k,l} u_l u_k.$$

We will pick a random Gaussian vector u. Then,

$$\begin{aligned} M_2 = (M - M_1)(u,u) &= \sum_i (\mathsf{E}\,(s_i^4) - 3) A^{(i)} \otimes A^{(i)} (A^{(i)} \cdot u)^2 \\ &= \sum_i (\mathsf{E}\,(s_i^4) - 3)(A^{(i)} \cdot u)^2 A^{(i)} \otimes A^{(i)} \\ &= A \begin{bmatrix} \mathsf{E}\,(s_i^4)(A^{(1)} \cdot u)^2 & & 0 \\ & \ddots & \\ 0 & & (\mathsf{E}\,(s_i^4) - 3)(A^{(n)} \cdot u)^2 \end{bmatrix} A^T \\ &= ADA^T \end{aligned}$$

Since u is random, with high probability, the nonzero entries of D will be distinct. Thus the eigenvectors of M_2 will be the columns of A (note that this is after making A an orthonormal matrix).

3.2 Fourier PCA and noisy ICA

Here we assume data is generated from the model

$$x = As + \eta,$$

where $\eta \sim N(\mu, \Sigma)$ is Gaussian noise with unknown mean μ and unknown covariance Σ. As before, the problem is to estimate A. To do this, we consider a different algorithm, first for the noise-free case.

Algorithm: Noisy ICA

1. Pick vectors $u, v \in \mathbf{R}^n$.
2. Compute weights $\alpha(x) = e^{u^T x}$, $\beta(x) = e^{v^T x}$.
3. Compute the covariances of the sample w.r.t. both weightings:

$$\tilde{\mu}_u = \frac{\mathsf{E}\,(e^{u^T x} x)}{\mathsf{E}\,(e^{u^T x})}, \qquad M_u = \frac{\mathsf{E}\,(e^{u^T x}(x - \tilde{\mu}_u)(x - \tilde{\mu}_u^T)}{\mathsf{E}\,(e^{u^T x})}$$

4. Output the eigenvectors of $M_u M_v^{-1}$.

Theorem 3.2. *The algorithm above recovers A to any desired accuracy, under the assumption that at most one component is Gaussian. The time and sample complexity of the algorithm are polynomial in $n, \sigma_{min}, 1/\epsilon$ and exponential in $k = \max_i k_i$ and for each component i, the index k_i is the smallest index at which the k_i'th cumulant of s_i is nonzero.*

Proof. We begin by computing the (i,j)-th entries of M_u,

$$\begin{aligned}
(M_u)_{i,j} &= \frac{\mathsf{E}\,(e^{u^T x}(x_i - \tilde{\mu}_i)(x_i - \tilde{\mu}_i)^T)}{\mathsf{E}\,(e^{u^T x})} \\
&= \frac{\mathsf{E}\,(e^{u^T x}(x_i x_j - \tilde{\mu}_i x_j - x_i \tilde{\mu}_j + \tilde{\mu}_i \tilde{\mu}_j)}{\mathsf{E}\,(e^{u^T x})} \\
&= \frac{\mathsf{E}\,(e^{u^T x}(x_i x_j)) - \tilde{\mu}_i \tilde{\mu}_j \mathsf{E}\,(e^{u^T x})}{\mathsf{E}\,(e^{u^T x})}
\end{aligned}$$

Next, we may rewrite $\tilde{\mu}$ in terms of A and $\bar{s}$,

$$\begin{aligned}
\tilde{\mu}_i &= \frac{\mathsf{E}\,(e^{u^T x} x_i)}{\mathsf{E}\,(e^{u^T x})} \\
\tilde{\mu} &= A\bar{s},
\end{aligned}$$

such that we can substitute for $\tilde{\mu}$ in M_u,

$$\begin{aligned} M_u &= \frac{\mathsf{E}\left(e^{u^T x}(xx^T)\right))}{\mathsf{E}\left(e^{u^T x}\right)} - \tilde{\mu}\tilde{\mu}^T \\ &= \frac{A\mathsf{E}\left(e^{u^T x}(ss^T)\right)A^T}{\mathsf{E}\left(e^{u^T As}\right)} - \tilde{\mu}\tilde{\mu}^T \\ &= \frac{A\mathsf{E}\left(e^{u^T x}(s-\bar{s})(s-\bar{s})^T\right)A^T}{\mathsf{E}\left(e^{u^T As}\right)} \\ &= A\begin{bmatrix} D_1 & & 0 \\ & \ddots & \\ 0 & & D_n \end{bmatrix}A^T \\ &= ADA^T, \end{aligned}$$

where the diagonal entries of the matrix D are defined as,

$$D_i = \frac{\mathsf{E}\left(e^{u^T x}(s-\bar{s})(s-\bar{s})^T\right)}{\mathsf{E}\left(e^{u^T As}\right)}.$$

Doing this for both u and v, we have

$$M_u M_v^{-1} = AD_u D_v^- 1 A^{-1}$$

whose eigenvectors are the columns of A. Note that $M_u M_v^{-1}$ is not symmetric in general and its eigenvectors need not be orthogonal. □

To adapt the above algorithm to handle Gaussian noise, we simply modify the last step to the following:

- Output the eigenvalues and eigenvectors of $(M_u - M)(M_v - M)^{-1}$

where M is the covariance matrix of the original matrix (with no weighting).

Exercise 3.2. *Show that the above variant of Fourier PCA recovers the columns of an ICA model $Ax + \eta$ for any unknown Gaussian noise η.*

3.3 Discussion

Chapter 4

Recovering Planted Structures in Random Graphs

Here we study the problem of analyzing data sampled from discrete generative models, such as planted cliques and partitions of random graphs.

4.1 Planted cliques in random graphs

$G_{n,p}$ denotes a family of random graphs, with n being the number of vertices in the graph and p being the probability of an edge existing between any (distinct) pair of vertices, also called the *edge density* of the graph. This is equivalent to independently filling up the upper triangle of the $n \times n$ adjacency matrix A with entries 1 with probabilty p and 0 with probability $1 - p$ (the diagonal has zero entries, and the lower triangle is a copy of the upper one as the adjacency matrix is symmetric). The graphs of most interest is the family $G_{n,1/2}$, and we usually refer to a graph sampled from this family when we talk about a random graph, without any other qualifiers.

4.1.1 Cliques in random graphs

The *maximum clique problem* is very well known to be NP-hard in general. In fact, even the following approximation version of the problem is NP-hard: to find a clique of size $OPT/n^{1-\epsilon}$ for any $\epsilon > 0$, where OPT represents the size of the maximum clique (the clique number) and n is the number of vertices in the graph, as usual. For a random graph, however, the situation is better understood. For instance, the following result about the clique number in $G_{n,1/2}$ is a standard exercise in the probabilistic method.

Exercise 4.1. *Prove that with high probability* $(1 - o(1))$*, the clique number of* $G_{n,1/2}$ *is* $(2 + o(1)) \log_2 n$*.*

[Hint: Use the first and second moments to prove that the number of cliques for $2 \log_2 n - c \to \infty$ for $c \neq o(1)$ and $2 \log_2 n + c \to 0$ for $c \neq o(1)$.]

The result as stated above is an existential one. The question arises: how does one find a large clique in such a graph? It makes sense to pick the vertices with the highest degrees, as these have a high probability of being in the clique. This strategy leads to the following algorithm.

Algorithm: Greedy-Clique

1. Define S to be the empty set and $H = G$.
2. While H is nonempty,
   - add the vertex with highest degree in H to S, and,
   - remove from H all the vertices not adjacent to every vertex in S.
3. Return S.

Proposition 4.1. *For* $G_{n,1/2}$*, the above algorithm finds a clique of size at least* $\log_2 n$ *with high probability.*

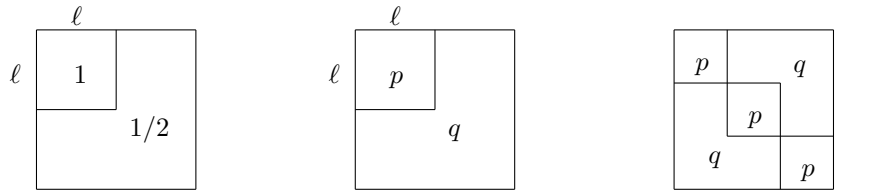
Exercise 4.2. *Prove Prop. 4.1.*

Exercise 4.3. *Consider the following simpler algorithm: pick a random vertex* v_1 *of* $G = G_{n,\frac{1}{2}}$*, then pick a random neighbor of* v_1*, and continue picking a random common neighbor of all vertices picked so far as long as possible. Let* k *be the number of vertices picked. Show that* $\mathsf{E}(k) \geq \log_2 n$ *and with high probability* $k \geq (1 - o(1)) \log_2 n$*.*

The size of the clique returned by the algorithm is only half of the expected clique number. It remains an open problem to understand the complexity of finding a $(1 + \epsilon) \log_2 n$ (for any $\epsilon > 0$) sized clique with high probability in polynomial time.

4.1.2 Planted clique

We now turn to the *planted clique problem*, which asks us to find an ℓ-vertex clique that has been "planted" in an otherwise random graph on n vertices, i.e., we choose some ℓ vertices among the n vertices of a random graph, and put in all the edges among those vertices. This generalizes to the *planted dense subgraph problem*, in which we have to recover a $H \leftarrow G_{\ell,p}$ that has been planted in a $G \leftarrow G_{n,q}$ for some $p > q$. This further generalizes to the *planted partition* problem in which the vertex set of $G \leftarrow G_{n,q}$ is partitioned into k



(a) Planted clique (b) Planted dense subgraph (c) Planted partitions

Figure 4.1: The adjacency matrices of the various generalizations of the planted clique problem. The figures inside the blocks represent the edge density of that block. The blocks have been shown to be contiguous for illustration purposes only.

pieces and subgraphs with edge density $p > q$ are planted into those partitions, with the requirement being to recover the partitions. The adjacency matrices of problem instances in each of the problem appear in Figure 4.1. A final generalization of this problem is the *planted data model problem*, where the vertex set is partitioned into k pieces and the edge density of the subgraph between the r^{th} and the s^{th} piece is denoted by p_{rs}, and we are required to recover the partitions and estimate the hidden p_{rs} values.

We now analyze the planted clique problem, where we have been provided with the adjacency matrix of a candidate graph containing a planted clique $S \subseteq [n]$, $|S| = \ell$. Given the adjacency matrix A of such a graph, the degree of any vertex i can be written as $d_i = \sum_{j=1}^{n} A_{ij}$ and it is easy to observe that $\mathsf{E}\ d_i = (n+\ell)/2$ if $i \in S$ and $n/2$ otherwise. By a simple application of Chernoff bounds, we get the following:

$$\mathbf{Pr}[d_i - \mathsf{E}\ d_i > t] \leq \exp -t^2/2n,$$

$$\mathbf{Pr}[d_i - \mathsf{E}\ d_i < -t] \leq \exp -t^2/2n.$$

For $t = 2\sqrt{n \log n}$, these probabilites become bounded by $1/n^2$. In other words, if $i \notin S$, then $\mathbf{Pr}[d_i > n/2 + 2\sqrt{n \log n}] \leq 1/n^2$ and the union bound give us

$$\mathbf{Pr}[\exists i \notin S \text{ s.t. } d_i > n/2 + 2\sqrt{n \log n}] \leq (n-\ell)/n^2 < 1/n.$$

Similarly, we also have that

$$\mathbf{Pr}[\exists i \in S \text{ s.t. } d_i < (n+\ell)/2 - 2\sqrt{n \log n}] \leq \ell/n^2 < 1/n.$$

So, if $\ell > 8\sqrt{n \log n}$, then the degrees of the vertices in the planted clique and the other vertices are well separated with high probability, and a greedy algorithm of picking the highest degree vertices as part of the clique would work. The question is, can we make this bound on ℓ smaller? In Section 4.2.1, we use spectral techinques to get a better bound on ℓ.

4.2 Full Independence and the Basic Spectral Algorithm

We return to the planted data model problem. Denoting by A the adjacency matrix of the input graph, the problem can be stated succinctly: given (one realization of) A, find $\mathsf{E}\, A$ the entry-wise expectation (since $\mathsf{E}\, A$ contains information on the partition as well as the p_{rs} values). To see why this is true, we recall the planted clique problem, where the adjacency matrix A has a $\ell \times \ell$ block of ones, and every other entry is 1 with probability $1/2$. In this scenario, $\mathsf{E}\, A$ has the same $\ell \times \ell$ block of ones, and every other entry is *exactly* $1/2$. Each row of the matrix is fully populated by $1/2$, except for the rows corresponding to the clique, which have ℓ ones in them, and it is very easy to distinguish the two cases.

We may view this as a mixture model. Denote by A the adjacency matrix of the graph. Each row $A_{(i)}$ is a point (with 0-1 coordinates) in $\mathbf{R}^n$ generated from a mixture of k probability distributions, where each component distribution generates the adjacency vectors of vertices in one part. It is of interest to cluster when the p_{rs} as well as their differences are small, i.e., $o(1)$. However, since the rows of A are 0-1 vectors, they are very "far" along coordinate directions (measured in standard deviations, say) from the means of the distributions. This is quite different from the case of a Gaussian (which has a very narrow tail). The fat tail is one of the crucial properties that makes the planted graph problem very different from the Gaussian mixture problem.

The basic tool which has been used to tackle heavy tails is the assumption of *full independence* which postulates that the edges of the graph are mutually independent random variables. This is indeed a natural conceptual off-shoot of random graphs. Now, under this assumption, the very rough outline of the spectral clustering algorithm is as follows: we are given A and wish to find the generative model $\mathsf{E}\, A$ which tells us the probabilities p_{rs} (and the parts). The matrix $A - \mathsf{E}\, A$ has random independent entries each with mean 0. There is a rich theory of random matrices where the generative model satisfies full independence and the following celebrated theorem was first stated qualitatively by the physicist Wigner.

Theorem 4.2. *Suppose A is a symmetric random matrix with independent (above-diagonal) entries each with standard deviation at most ν and bounded in absolute value by 1. Then, with high probability, the largest eigenvalue of $A - \mathsf{E}\, A$ is at most $(2 + o(1))\nu\sqrt{n}$.*

The strength of this Theorem is seen from the fact that each row of $A - \mathsf{E}\, A$ is of length $O(\nu\sqrt{n})$, so the Theorem asserts that the top eigenvalue amounts only to the length of a constant number of rows; i.e., there is almost no correlation among the rows (since the top eigenvalue $= \max_{|x|=1} \|(A - \mathsf{E}\, A)x\|$ and hence the higher the correlation of the rows in some direction x, the higher its value).

Thus one gets with high probability an upper bound on the spectral norm

of $A - EA$:

$$\|A - \mathsf{E}\ A\| \leq c\nu\sqrt{n}.^1$$

Now an upper bound on the Frobenius norm $\|A - \mathsf{E}\ A\|_F$ follows from the following basic lemma.

Lemma 4.3. *Suppose A, B are $m \times n$ matrices with rank$(B) = k$. If $\hat{A}$ is the best rank k approximation to A, then*

$$\|\hat{A} - B\|_F^2 \leq 8k\|A - B\|^2.$$

Proof. Since $\hat{A} - B$ has rank at most $2k$, and $\hat{A}$ is the best rank k approximation of A, we have,

$$\begin{aligned} \|\hat{A} - B\|_F^2 &\leq 2k\|\hat{A} - B\|_2^2 \\ &\leq 2k\left(\|\hat{A} - A\|_2 + \|A - B\|_2\right)^2 \\ &\leq 8k\|A - B\|_2^2. \end{aligned}$$

□

Exercise 4.4. *Improve the bound in Lemma 4.3 from $8k$ to $5k$.*

4.2.1 Finding planted cliques

In this Section, we see how to apply Theorem 4.2 to recover the planted clique in a random graph $G(n, 1/2)$ with only a $\Omega(\sqrt{n})$ lower bound on ℓ. We consider the following simple spectral algorithm.

Algorithm: Spectral-Clique

1. Let A be the $1/-1$ adjacency matrix of the input graph (say 1 for an edge and -1 for a nonedge).
2. Find the top eigenvector v of A.
3. Let S be the subset of ℓ vertices of largest magnitude in v.
4. Output all vertices whose degree in S is at least $7\ell/8$.

Theorem 4.4. *For a planted clique of size $\ell \geq 20\sqrt{n}$, with high probability, Algorithm Spectral-Clique recovers exactly the planted clique.*

Proof. We apply Lemma 4.3 (and the improvement in Exercise 4.4) with $k = 1$ and $B = \mathsf{E}\ A$, to get

$$\|\hat{A} - \mathsf{E}\ A\|_F^2 \leq 5\|A - \mathsf{E}\ A\|_2^2.$$

[1] We use the convention that c refers to a constant. For example, the statement $a \leq (cp)^{cp}$ will mean there exist constants c_1, c_2 such that $a \leq (c_1 p)^{c_2 p}$.

From Theorem 4.2, with $\nu = 1$, with high probability this is bounded as

$$\|\hat{A} - \mathsf{E}\ A\|_F^2 \leq 25n.$$

So for a random row i,

$$\mathsf{E}\ \|\hat{A}_{(i)} - \mathsf{E}\ A_{(i)}\|^2 \leq 25.$$

Therefore, using Markov's inequality,

$$\mathbf{Pr}(\|\hat{A}_{(i)} - \mathsf{E}\ A_{(i)}\|^2 \leq 25\epsilon\sqrt{n}) \leq \frac{\mathsf{E}\ \|\hat{A}_{(i)} - \mathsf{E}\ A_{(i)}\|^2}{25\epsilon\sqrt{n}} \leq \frac{1}{\epsilon\sqrt{n}}.$$

We have for all but $\sqrt{n}/\epsilon$ rows,

$$\|\hat{A}_{(i)} - \mathsf{E}\ A_{(i)}\|^2 \leq 25\epsilon\sqrt{n}. \tag{4.1}$$

If an edge is not in the planted clique, the corresponding entry in A is 1 with probability $1/2$ and -1 with probability $1/2$. So the expectation of such entries is zero. If the vertex i is in the planted clique, $\mathsf{E}\ A_{(i)}$ is the indicator vector of the clique, so $\|\mathsf{E}\ A\| = \sqrt{\ell}$. For rows satisfying (4.1), by the triangle inequality,

$$\|\hat{A}_{(i)}\| \geq \|\mathsf{E}\ A_{(i)}\| - \|\hat{A}_{(i)} - \mathsf{E}\ A_{(i)}\| \geq \sqrt{\ell} - \sqrt{25\epsilon\sqrt{n}}.$$

If the vertex i is not in the planted clique, $\mathsf{E}\ A_{(i)}$ is all zeros. Then for rows satisfying (4.1),

$$\|\hat{A}_{(i)}\| = \|\hat{A}_{(i)} - \mathsf{E}\ A_{(i)}\| \leq \sqrt{25\epsilon\sqrt{n}}.$$

By setting ϵ, we get that for all but $n - (\sqrt{n}/\epsilon)$ rows, the component in the eigenvector v of a vertices in the clique will be higher than the component of vertices outside the planted clique. To achieve this, we need that the length of $\hat{A}_{(i)} - \mathsf{E}\ A_{(i)}$ for vertices i outside the clique is smaller than that of vertices in the clique (for rows satisfying (4.1), i.e.,

$$\sqrt{\ell} - \sqrt{25\epsilon\sqrt{n}} \geq \sqrt{25\epsilon\sqrt{n}}.$$

So $\ell > \max\{100\epsilon\sqrt{n}, 8\sqrt{n}/\epsilon\}$ suffices, and we can set $\epsilon = 2/5$ and $\ell = 40\sqrt{n}$. Thus, the algorithm finds

$$\ell - \frac{\sqrt{n}}{\epsilon} \geq \frac{7\ell}{8}$$

vertices of the clique in the ℓ largest magnitude entries. Because of our careful choice of parameters, if some clique vertices are not found in S, they will be found in Step 4 of the algorithm. By an application of a Chernoff Bound, we can bound the probability of the degree of a non-clique vertex i in the clique C being at least $3\ell/4$:

$$\mathbf{Pr}[\deg_C(i) \geq \ell/2 + t] \leq e^{-t^2/2\ell} \implies \mathbf{Pr}[\deg_C(i) \geq 3\ell/4] \leq e^{-\ell/32} \leq e^{-20\sqrt{2n}/32}.$$

The total probability of a non-clique vertex being included in Step 4 of the algorithm would be bounded by the union bound of the above probability over all the non-clique vertices, and thus the failure probability (failure being the event of a non-clique vertex being included in S) is bounded above by $ne^{-20\sqrt{2n}/32}$, which is negligibly small. Therefore with this failure probability, the degree of any non-clique vertex is less than $3\ell/4 + \ell/8 = 7\ell/8$ and no such vertex will be included in the final output. [Note that we cannot directly bound the degree in the set S since this set is not fixed prior to examining the graph.] □

Exercise 4.5. *Suppose a clique of size ℓ is planted in the random graph $G_{n,p}$ where every edge not in the clique is chosen independently with probability p. Generalize the spectral algorithm for finding the planted clique and derive a bound on the size of the clique (in terms of p and n) that can be found by the algorithm* **whp** *. [Hint: you might need the result of Exercise 4.8.]*

4.3 Proof of the spectral norm bound

Here we prove Wigner's theorem (Thm. 4.2) for matrices with random ± 1 entries. The proof is probabilistic, unlike the proof of the general case for symmetric distributions. The proof has two main steps. In the first step, we use a discretization (due to Kahn and Szemerédi) to reduce from all unit vectors to a finite set of lattice points. The second step is a Chernoff bound working with fixed vectors belonging to the lattice.

Let $\mathcal{L}$ be the lattice $\left(\frac{1}{r\sqrt{n}}\mathbb{Z}\right)^n$. The diagonal length of its basic parallelepiped is $diag(\mathcal{L}) = 1/r$.

Lemma 4.5. *Any vector $u \in \boldsymbol{R}^n$ with $\|u\| = 1$ can be written as*

$$u = \lim_{N\to\infty} \sum_{i=0}^{N} \left(\frac{1}{2r}\right)^i u_i$$

where

$$\|u_i\| \leq 1 + \frac{1}{2r},\ \forall\ i \geq 0.$$

and $u_i \in \mathcal{L},\ \forall\ i \geq 0$.

Proof. Given $u \in \boldsymbol{R}^n$ with $\|u\| = 1$, we pick $u_0 \in \mathcal{L}$ to be its nearest lattice point. Therefore,

$$\|u_0\| \leq 1 + \frac{diag(\mathcal{L})}{2} = 1 + \frac{1}{2r}$$

Now $(u - u_0)$ belongs to some basic parallelepiped of $\mathcal{L}$ and therefore $\|u - u_0\| \leq 1/2r$. Consider the finer lattice $\mathcal{L}/2r = \{x/2r\ :\ x \in \mathcal{L}\}$, and pick $u_1/2r$ to be the point nearest to $(u - u_0)$ in $\mathcal{L}/2r$. Therefore,

$$\|\frac{u_1}{2r}\| \leq \|u - u_0\| + \frac{diag(\mathcal{L}/2r)}{2} \leq \frac{1}{2r} + \frac{1}{(2r)^2} \implies \|u_1\| \leq 1 + \frac{1}{2r}$$

and

$$\|u - u_0 - \frac{1}{2r}u_1\| \leq \frac{1}{(2r)^2}$$

Continuing in this manner we pick $u_k/(2r)^k$ as the point nearest to $\left(u - \sum_{i=0}^{k-1}(1/2r)^i u_i\right)$ in the finer lattice $\mathcal{L}/(2r)^k = \left\{x/(2r)^k \ : \ x \in \mathcal{L}\right\}$. Therefore, we have

$$\|\frac{u_k}{(2r)^k}\| \leq \|u - \sum_{i=0}^{k-1}\left(\frac{1}{2r}\right)^i u_i\| + \frac{diag(\mathcal{L}/(2r)^k)}{2} \leq \frac{1}{(2r)^k} + \frac{1}{(2r)^{k+1}}$$

$$\implies \|u_k\| \leq 1 + \frac{1}{2r} \implies \|u - \sum_{i=0}^{k}\left(\frac{1}{2r}\right)^i u_i\| \leq \frac{1}{(2r)^{k+1}} \longrightarrow 0$$

That completes the proof. □

Now using Lemma 4.5, we will show that it suffices to consider only the lattice vectors in $\mathcal{L} \cap \mathbb{B}(\bar{0}, 1 + 1/2r)$ instead of all unit vectors in order to bound $\lambda(A)$. Indeed, this bound holds for the spectral norm of a tensor.

Proposition 4.6. *For any matrix A,*

$$\lambda(A) \leq \ \left(\frac{2r}{2r-1}\right)^2 \ \left(\sup_{u,v \in \mathcal{L} \cap \mathbb{B}(\bar{0}, 1+\frac{1}{2r})} \left|u^T A v\right|\right)$$

Proof. From Lemma 4.5, we can write any u with $\|u\| = 1$ as

$$u = \lim_{N \to \infty} \sum_{i=0}^{N}\left(\frac{1}{2r}\right)^i u_i$$

where $u_i \in \mathcal{L} \cap \mathbb{B}(\bar{0}, 1 + 1/2r), \ \forall \ i$. We similarly define v_j. Since $u^T A v$ is a continuous function, we can write

$$\begin{aligned}
\left|u^T A v\right| &= \lim_{N \to \infty} \left| \left(\sum_{i=0}^{N}\left(\frac{1}{2r}\right)^i u_i\right)^T A \sum_{j=0}^{\infty}\left(\frac{1}{2r}\right)^j v_j \right| \\
&\leq \left(\sum_{i=0}^{\infty}\left(\frac{1}{2r}\right)^i\right)^2 \sup_{u,v \in \mathcal{L} \cap \mathbb{B}(\bar{0}, 1+\frac{1}{2r})} \left|u^T A v\right| \\
&\leq \left(\frac{2r}{2r-1}\right)^2 \sup_{u,v \in \mathcal{L} \cap \mathbb{B}(\bar{0}, 1+\frac{1}{2r})} \left|u^T A v\right|
\end{aligned}$$

which proves the proposition. □

We also show that the number of r vectors $u \in \mathcal{L} \cap \mathbb{B}(\bar{0}, 1 + 1/2r)$ that we need to consider is at most $(2r)^n$.

Lemma 4.7. *The number of lattice points in $\mathcal{L} \cap \mathbb{B}(\bar{0}, 1+1/2r)$ is at most $(2r)^n$.*

Proof. We can consider disjoint hypercubes of size $1/r\sqrt{n}$ centered at each of these lattice points. Each hypercube has volume $(r\sqrt{n})^{-n}$, and their union is contained in $\mathbb{B}(\bar{0}, 1 + 2/r)$. Hence,

$$\begin{aligned} |\mathcal{L} \cap \mathbb{B}(\bar{0}, 1 + 1/2r)| &\leq \frac{\text{Vol}\left(\mathbb{B}(\bar{0}, 1 + 2/r)\right)}{(r\sqrt{n})^{-n}} \\ &\leq \frac{2\pi^{n/2}(1 + \frac{2}{r})^n r^n n^{n/2}}{\Gamma(n/2)} \\ &\leq (2r)^n \end{aligned}$$

□

The following Chernoff bound will be used.

Exercise 4.6. *Let $X_1, X_2, \ldots, X_m$ be independent random variables, $X = \sum_{i=1}^m X_i$, where each X_i is a_i with probability $1/2$ and $-a_i$ with probability $1/2$. Let $\sigma^2 = \sum_{i=1}^m a_i^2$. Then, for $t > 0$,*

$$\mathbf{Pr}\left(|X| \geq t\sigma\right) \leq 2e^{-t^2/2}$$

Now we can prove the spectral norm bound for a matrix with random ± 1 entries.

Proof. Consider fixed $u, v \in \mathcal{L} \cap \mathbb{B}(\bar{0}, 1+1/2r)$. For $I = (i, j)$, define a two-valued random variable

$$X_I = A_{ij} u_i v_j.$$

Thus $a_I = u_i v_j$, $X = \sum_I X_I = u^T A v$, and

$$\sigma^2 = \sum_I a_I^2 = \|u\|^2 \|v\|^2 \leq \left(\frac{2r+1}{2r}\right)^4.$$

So using $t = 4\sqrt{n}\sigma$ in the Chernoff bound 4.6,

$$\mathbf{Pr}\left(\left|u^T A v\right| \geq 4\sqrt{n} \cdot \sigma\right) \leq 2e^{-8n}.$$

According to Lemma 4.7, there are at most $(2r)^{2n}$ ways of picking $u, v \in \mathcal{L} \cap \mathbb{B}(\bar{0}, 1 + 1/2r)$. so we can use union bound to get

$$\mathbf{Pr}\left(\sup_{u,v \in \mathcal{L} \cap \mathbb{B}(\bar{0}, 1+\frac{1}{2r})} \left|u^T A v\right| \geq 4\sqrt{n}\sigma\right) \leq (2r)^{2n}(e)^{-8n} \leq e^{-5n}$$

for $r = 2$. And finally using Proposition 4.6 and the facts that for our choice of r, $\sigma \leq 25/16$ and $(2r/2r-1)^2 = 16/9$, we have

$$\mathbf{Pr}\left(\lambda(A) \geq 11\sqrt{n}\right) \leq e^{-5n}.$$

This completes the proof. □

The above bound can be extended in the following ways.

Exercise 4.7. *Let A be an $n \times n \times \ldots \times n$ r-dimensional array with real entries. Its spectral norm $\lambda(A)$ is defined as*

$$\lambda(A) = \sup_{\|u^{(1)}\|=\|u^{(2)}\|=\ldots=\|u^{(r)}\|=1} \left| A\left(u^{(1)}, u^{(2)}, \ldots, u^{(r)}\right)\right|,$$

where $A\left(u^{(1)}, u^{(2)}, \ldots, u^{(r)}\right) = \sum_{i_1,i_2,\ldots,i_r} A_{(i_1,i_2,\ldots,i_r)}\ u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(r)}_{i_r}$. Suppose each entry of A is 1 or -1 with equal probability. Show that whp,

$$\lambda(A) = O(\sqrt{n} r \log r). \tag{4.2}$$

Exercise 4.8. *Let each entries of an $n \times n$ matrix A be chosen independently to be 1 with probability p and 0 with probability $1-p$. Give a bound on the spectral norm of $A - \mathsf{E}\ A$.*

.

4.4 Planted partitions

For the general planted partition problem, we use Lemma 4.3 and Theorem 4.2 with $B = \mathsf{E}\ A$ and ν equal to the maximum standard deviation of any row of A in any direction. We can find the SVD of A to get $\hat{A}$. By the above, we have that whp,

$$\|\hat{A} - \mathsf{E}\ A\|_F^2 \leq c\nu^2 nk$$

Let ϵ be a positive real $< 1/(10k)$. Then for all but a small fraction of the rows, we find the vectors $(\mathsf{E}\ A)_{(i)}$ within error $c\nu\sqrt{k}$; i.e., for all but ϵn of the rows of A, we have (whp)

$$|\hat{A}_{(i)} - \mathsf{E}\ A_{(i)}| \leq c\nu\sqrt{\frac{k}{\epsilon}}.$$

Let G be the set of rows of A satisfying this condition.

Now, we assume a **separation condition** between the centers μ_r, μ_s of the component distributions $r \neq s$ (as in the case of Gaussian mixtures):

$$\|\mu_r - \mu_s\| \geq \Delta = 20c\nu\sqrt{\frac{k}{\epsilon}}.$$

We note that Δ depends only on k and not on n (recall that $k << n$). In general, a point $A_{(i)}$ may be at distance $O(\sqrt{n}\nu)$ from the center of its distribution which is much larger than Δ.

It follows that points in G are at distance at most $\Delta/20$ from their correct centers and at least 10 times this distance from any other center. Thus, each point in G is at distance at most $\Delta/10$ from every other point in G in its own part and at distance at least $\Delta/2$ from each point in G in a different part. We use this to cluster most points correctly as follows:

Pick at random a set of k points from the set of projected rows by picking each one uniformly at random from among those at distance at least $9c\nu\sqrt{k/\epsilon}$ from the ones already picked. This yields with high probability k good points one each from each cluster, assuming $\epsilon < 1/(10k)$. We define k clusters, each consisting of the points at distance at most $\Delta/5$ from each of the k points picked.

After this, all known algorithms resort to a **clean-up** phase where the wrongly clustered vertices are reclassified correctly. The clean-up phase is often technically very involved and forces stricter (and awkward) separation conditions. Here we outline very briefly a possible clean-up procedure.

To correctly cluster a particular i : Let B be the matrix obtained from A with row i deleted. We will see that this is to avoid any conditioning. As above, we have with high probability

$$||B - \mathsf{E}\ B|| \leq c\nu\sqrt{n}.$$

We have to ensure that the failure probability is low enough so that no failure occurs for any of the n i 's. Let $\hat{B}$ be the rank k approximation to B. As above, except for ϵn "bad points", we have

$$||\hat{B}_{(j)} - \mathsf{E}\ B_{(j)}|| \leq \Delta/20.$$

We again pick k points from among the rows of $\hat{B}$ which with high probability are good points, each from a different cluster. To avoid unnecessary notation, call these k points $\hat{B}_{(1)}, \hat{B}_{(2)}, \ldots \hat{B}_{(k)}$ and suppose $\hat{B}_{(1)}$ is within distance $\Delta/10$ of the center of the distribution from which $A_{(i)}$ is picked. Now imagine that we have done all this before picking $A_{(i)}$; we may do so since $A_{(i)}$ is independent of all this. Now consider the projection of $A_{(i)}$ onto the k dimensional space spanned by $\hat{B}_{(1)}, \hat{B}_{(2)}, \ldots \hat{B}_{(k)}$. Under reasonable assumptions, we can show that in this projection $A_{(i)}$ is closer to $\hat{B}_{(1)}$ than to $\hat{B}_{(2)}, \hat{B}_{(3)}, \ldots \hat{B}_{(k)}$. [The assumptions are to the effect that $O(n)$ coordinates of each center are non-zero. This is to avoid the situation when a distribution is based only on $o(n)$ or in the extreme case just $O(1)$ coordinates; such "unbalanced" distributions have fat tails.] Assuming this, we now can conclude that i is in the same cluster as 1 (since we know all of $\hat{B}_{(1)}, \ldots \hat{B}_{(k)}$ and $\hat{B}_{(i)}$. We may repeat the process of picking k near-centers from $\hat{B}$ $O(1)$ times to get a number of points which belong to the same cluster as i. The whole process has to be repeated with each i and one can complete the clustering by using all the information gathered on pairs of points in the same cluster.

4.5 Beyond full independence

Motivated by data mining and information retrieval applications, the question of inferring the generative model from the data has moved from random graphs to matrices where the rows represent *features* and the columns represent *objects* and the (i, j)'th entry is the value of feature i for object j. Two salient examples

are product-consumer matrices, widely used in recommendation systems, and term-document matrices. Recent work on recommendation systems uses the full independence assumptions, so that the procedures described earlier can be carried out.

In both these examples, as well as others, it is easy to argue that the full independence assumption is too strong. While one may reasonably assume that consumers function independently, a particular consumer might not decide independently on each product — at the minimum, he/she might have constraints on the total budget (and perhaps some others) which results in correlations of the products bought. Similarly, while documents in a collection may be drawn, say, from a mixture, the terms that occur in each document are clearly not chosen independently. This points to the following model and problem:

In a generative model where a collection of objects is chosen from a mixture distribution, infer the model, given the objects. So the columns of the matrix are independent vector-valued random variables. The entries in a column are not independent.

The crucial tool we need is a Wigner-type theorem in this situation. Such a theorem follows from results in functional analysis and probability theory. But these are not readily accessible, so we will present a self-contained proof of the following theorem (including the important classical technique of decoupling used in the proof). The Theorem states that a Wigner-type bound holds with just the limited independence assumption (that the columns are independent) if we allow some extra logarithmic factors (see the corollary below).

A definition will be useful: for a vector-valued random variable Y, we define the variance of Y denoted $\mathsf{Var}\,(Y)$ as the maximum variance of the real-valued random variable $v \cdot Y$, where the maximum is taken over all unit length vectors v. I.e., it is the maximum variance of Y in any direction. It is easy to see that it is the maximum eigenvalue of the covariance matrix of Y:

$$\mathsf{Var}\,(Y) = \|\mathsf{E}\, YY^T\|.$$

We first need a well-known fact (see for example, [Bha97], IV.31). Throughout, X, X_i will represent matrices. They may be rectangular and the entries are assumed to be real numbers.

Proposition 4.8. *For $p > 0$, $||X||_p = (Tr(XX^T)^{p/2})^{1/p} = (Tr(X^TX)^{p/2})^{1/p}$ is a norm[2] (called a Schatten $p-$norm). Hence it is a convex function of the entries of the matrix X.*

Exercise 4.9. *Show that $||X||_p \leq ||X||_q$ for $p \geq q$.*

Recall that the trace of a matrix is the sum of its eigenvalues and $||X||_\infty$ (also denoted $||X||$) is the spectral norm.

[2]Since XX^T is positive semi-definite, $(XX^T)^{p/2}$ is well-defined. Namely if $XX^T = \sum_i \lambda_i u^{(i)} u^{(i)^T}$ is the spectral decomposition, then $(XX^T)^{p/2} = \sum_i \lambda_i^{p/2} u^{(i)} u^{(i)^T}$.

Theorem 4.9. *Suppose A is an $m \times n$ matrix with independent vector-valued random variables as its columns. Suppose the variance of $A^{(i)}$ is ν_i^2. Then for any p which is a power of 2, we have*

$$\mathsf{E}\,||A - \mathsf{E}\,A||_p^p \leq (cp)^{cp}\left(n\mathsf{E}\,\max_i |A^{(i)} - \mathsf{E}\,A^{(i)}|^p + n^{(p/2)+1}\sum_i \nu_i^p\right).$$

Corollary 4.10. *Suppose with probability at least $1-\delta$, we have $\|A^{(i)} - \mathsf{E}\,A^{(i)}\| \leq M$ for all i, then for all $t > 0$,*

$$\mathbf{Pr}\left(\|A - \mathsf{E}\,A\| \geq (c\log n)^c t(M + \sqrt{n}\max_i \nu_i)\right) \leq \delta + \frac{1}{n^{\log t/10}}.$$

Proof. We have that, for all i,

$$\|A^{(i)} - \mathsf{E}\,A^{(i)}\| \leq M.$$

Apply the Theorem with $p = 2\log n + 1$. □

In the full independent case, we can take $M = \nu\sqrt{n}\log n$ and δ very small. So, we get Wigner-type result in that case, but with additional log factors.

4.5.1 Sums of matrix-valued random variables

Throughout this section, $X_1, X_2, \ldots X_n, X_1', X_2', \ldots X_n'$ are independent matrix-valued random variables with X_i' having the same distribution as X_i and $\mathsf{E}\,X_i = 0$ for $i = 1, 2, \ldots n$. Let p be a positive even integer.

Note that

$$\|X\|_2^2 = \mathrm{Tr}XX^T = \|X\|_F^2.$$

We need the following well-known generalization of Hölder's inequality to matrices.

Proposition 4.11. *Suppose $A_1, A_2, \ldots A_m$ are matrices (of dimensions so that their product is well-defined). We have for any positive reals $r_1, r_2, \ldots r_m$ with $\sum_{i=1}^m \frac{1}{r_i} = 1$:*

$$\|A_1 A_2 \ldots A_m\|_p \leq \|A_1\|_{pr_1}\|A_2\|_{pr_2}\ldots\|A_m\|_{pr_m}.$$

Theorem 4.12. *[Square-Form Theorem] Suppose X_i are as above and p is a power of 2. Then,*

$$\mathsf{E}\,\|\sum_{i=1}^n X_i\|_p^p \leq p^{7p}10^p\left(\mathsf{E}\,\|\sum_{i=1}^n X_i X_i^T\|_{p/2}^{p/2} + \mathsf{E}\,\|\sum_{i=1}^n X_i^T X_i\|_{p/2}^{p/2}\right).$$

Proof. By induction on p. For $p = 2$, we have

$$\mathsf{E}\,||\sum_i X_i||_2^2 = \mathsf{E}\,\mathrm{Tr}\sum_{i,j} X_i X_j^T = \mathrm{Tr}\mathsf{E}\,\sum_{i,j} X_i X_j^T = \mathsf{E}\,\mathrm{Tr}\sum_i X_i X_i^T,$$

since

$$\mathsf{E}\ X_i X_j^T = \mathsf{E}\ X_i \mathsf{E}\ X_j^T = 0$$

for $i \neq j$. Now since $\sum_i X_i X_i^T$ is p.s.d., all its eigenvalues are non-negative and so,

$$\mathrm{Tr} \sum_i X_i X_i^T = \| \sum_i X_i X_i^T \|_1^1$$

proving the case of $p = 2$.

Now for general p,

$$\mathsf{E} \| \sum_{i=1}^n X_i \|_p^p \quad \leq \quad \mathsf{E} \| \sum_i X_i \sum_j X_j^T \|_{p/2}^{p/2} 2^{p/2} \mathsf{E} \| \sum_i X_i X_i^T \|_{p/2}^{p/2} + 8^{p/2} \mathsf{E} \| \sum_i X_i Y^T \|_{p/2}^{p/2}.$$

where $Y = \sum_j X_j'$ and we have used decoupling as in Lemma 4.13 below. Note that

$$\mathsf{E}\ X_i Y^T = \mathsf{E}\ X_i \mathsf{E}\ Y^T = 0$$

since X_i, Y are independent. We now use induction (with the notation that $[X_1|X_2|\dots X_n]$ denotes the matrix with $X_1, X_2, \dots X_n$ written in that order)and D is the matrix with n diagonal blocks, each equal to Y^T.

$$\begin{aligned}
& \mathsf{E} \,||\sum_i X_i Y^T||_{p/2}^{p/2} \\
\leq\ & 10^{p/2}(p/2)^{3.5p} \left(\mathsf{E} \,||\sum_i (Y X_i^T X_i Y^T)||_{p/4}^{p/4} + \mathsf{E} \,||\sum_i (X_i Y^T Y X_i^T)||_{p/4}^{p/4} \right) \\
=\ & 10^{p/2}(p/2)^{3.5p} \left(\mathsf{E} \,||Y \left(\sum_i (X_i^T X_i) \right) Y^T||_{p/4}^{p/4} + \mathsf{E} \,||[X_1|X_2|\dots|X_n] D^T D [X_1|X_2|\dots X_n]^T)||_{p/4}^{p/4} \right) \\
\leq\ & 10^{p/2}(p/2)^{3.5p} \left(\mathsf{E} \,||Y||_p^{p/2} ||\sum_i X_i^T X_i||_{p/2}^{p/4} + \mathsf{E} \,||Y||_p^{p/2} ||[X_1|X_2|\dots X_n]||_p^{p/2} \right) \\
& \text{using Prop. 4.11} \\
\leq\ & 10^{p/2}(p/2)^{3.5p} \mathsf{E} \,||Y||_p^{p/2} \left(||\sum_i X_i^T X_i||_{p/2}^{p/4} + ||\sum_i X_i X_i^T||_{p/2}^{p/4} \right) \\
\leq\ & 2 \cdot 10^{p/2}(p/2)^{3.5p} \left(\mathsf{E} \,||\sum_i X_i||_p^p \right)^{1/2} \left(\mathsf{E} \,||\sum_i X_i X_i^T||_{p/2}^{p/2} + \mathsf{E} \,||\sum_i X_i^T X_i||_{p/2}^{p/2} \right)^{1/2}
\end{aligned}$$

where the fourth line is a bit tricky: if X_i are $m \times q$ and $m_0 = \min(m, q)$, we

use

$$\begin{aligned} & \|[X_1|X_2|\dots|X_n]D^T D[X_1|X_2|\dots X_n]^T)\|_{p/4}^{p/4} \\ = & \sum_{t=1}^{m_0} \sigma_t^{p/2}([X_1|X_2|\dots|X_n]D^T) \\ \leq & (\sum_{t=1}^{m_0} \sigma_t^p[X_1|X_2|\dots|X_n])^{1/2}(\sum_{t=1}^{m_0} \sigma_t(D)^p)^{1/2} \\ \leq & \|Y\|_p^{p/2}\|[X_1|X_2|\dots|X_n]\|_p^{p/2}. \end{aligned}$$

the last using Jensen's inequality and the fact that $Y, \sum_i X_i$ have same distribution. Letting

$$x = \sqrt{\mathsf{E}\,||\sum_i X_i||_p^p} \quad \text{and} \quad b = \mathsf{E}\,||\sum_i X_i X_i^T||_{p/2}^{p/2} + \mathsf{E}\,||\sum_i X_i^T X_i||_{p/2}^{p/2},$$

this yields the following quadratic inequality for x:

$$x^2 \leq 2^{p/2}b + 2 \cdot 8^{p/2}10^{p/2}(p/2)^{3.5p}\sqrt{b}\, x$$

which implies that

$$x^2 \leq 10^p p^{7p} b,$$

completing the inductive proof. □

4.5.2 Decoupling

We now introduce a beautiful technique developed by probabilists and functional analysts called *decoupling* which helps get rid of some dependencies between random variables, making the analysis easier in many contexts. (See for example [**?**]). Decoupling looks like sleight of hand, but it is quite useful. It has been extensively applied in similar contexts.

Suppose f is any convex function from the set of matrices to non-negative reals with $f(A) = f(-A)$ and satisfying the condition that there is some $p > 0$ such that

$$f(A + B) \leq 2^p(f(A) + f(B)).$$

Typical examples of f will be p'th powers of norms.

Lemma 4.13. *Let $X_1, X_2, \dots X_n, X_1', X_2', \dots X_n'$ be independent matrix-valued random variables with X_i' having the same distribution as X_i and $\mathsf{E}\, X_i = 0$ for $i = 1, 2, \dots n$. Then, for any even convex function f,*

$$\mathsf{E}\, f\left(\sum_i X_i \sum_j X_j^T\right) \leq 8^p \mathsf{E}\, f\left(\sum_i X_i \sum_j X_j'^T\right) + 2^p \mathsf{E}\, f\left(\sum_i X_i X_i^T\right).$$

The point of the Lemma is that the first term on the RHS is easier to handle than the LHS, since now X_i', X_i are independent.

Proof. We let $Y_i = \{X_i, X_i'\}$ (the set (without order) of the two elements X_i, X_i') and $Y = (Y_1, Y_2, \ldots Y_n)$. We define random variables $Z_1, Z_2, \ldots Z_n, Z_1', Z_2', \ldots Z_n'$ as follows : for each i, independently, with probability 1/2 each, we let $(Z_i, Z_i') = (X_i, X_i')$ or $(Z_i, Z_i') = (X_i', X_i)$. Then, we clearly have

$$\begin{aligned}
\mathsf{E}\,(Z_i Z_j'^T | Y_i) &= \frac{1}{4}(X_i X_j^T + X_i' X_j^T + X_i X_j'^T + X_i' X_j'^T) \text{for } i \neq j \\
\mathsf{E}\,(Z_i Z_i'^T | Y_i) &= \frac{1}{2}(X_i X_i'^T + X_i' X_i^T).
\end{aligned}$$

$$\begin{aligned}
& \mathsf{E}\, f\left(\sum_i X_i \sum_j X_j^T\right) \\
\leq\ & 2^p \mathsf{E}\, f(\sum_i X_i X_i^T) + 2^p \mathsf{E}\, f(\sum_{i \neq j} X_i X_j^T) \\
\leq\ & 2^p \mathsf{E}\, f(\sum_i X_i X_i^T) + \\
& 2^p \mathsf{E}\, f\left(\sum_{i \neq j} \left(X_i X_j^T + \mathsf{E}\, X_i X_j'^T + \mathsf{E}\, X_i' X_j^T + \mathsf{E}\, X_i' X_j'^T\right) + 2\sum_i (\mathsf{E}\, X_i X_i'^T + \mathsf{E}\, X_i' X_i^T)\right) \\
\leq\ & 2^p \mathsf{E}\, f(\sum_i X_i X_i^T) + \\
& 2^p \mathsf{E}\, f\left(\sum_{i \neq j} \left(X_i X_j^T + X_i X_j'^T + X_i' X_j^T + X_i' X_j'^T\right) + 2\sum_i (X_i X_i'^T + X_i' X_i^T)\right) \\
& \text{using Jensen and convexity of } f, \text{ so } f(\mathsf{E}\, X) \leq \mathsf{E}\, f(X) \\
\leq\ & 2^p \mathsf{E}\, f(\sum_i X_i X_i^T) + 8^p \mathsf{E}\, f((\sum_i Z_i \sum_j Z_j'^T) | Y) \\
\leq\ & 2^p \mathsf{E}\, f(\sum_i X_i X_i^T) + 8^p \mathsf{E}\, f(\sum_i Z_i \sum_j Z_j'^T) \text{ using Jensen again .}
\end{aligned}$$

Now, the Lemma follows noting that $\{(Z_i, Z_j') : i = 1, 2, \ldots n\}$, and $\{(X_i, X_j') : i = 1, 2, \ldots n\}$ have the same joint distributions. □

4.5.3 Proof of the spectral bound with limited independence

We need another standard and useful trick:

Lemma 4.14. *Suppose $X_1, X_2, \ldots X_n$ are independent matrix-valued random variables with $EX_i = 0$. Let $\zeta = (\zeta_1, \zeta_2, \ldots \zeta_n)$, be a set of independent variables taking on values ± 1 with probability 1/2 each* [3]*, which are also independent of*

[3] These are referred to as Rademacher random variables in the literature

$X = (X_1, X_2, \ldots X_n)$. *We have*

$$E_X||\sum_i X_i||_p^p \leq 2^{p+1} E_{X,\zeta}||\sum_i \zeta_i X_i||.$$

Proof. Let $X' = (X_1', X_2', \ldots X_n')$ be a set of independent r.v. (independent of X_i, ζ_i).

$$\begin{aligned}
& \mathsf{E}_X||\sum_i X_i||_p^p \\
= \; & \mathsf{E}_X||\sum_i (X_i - \mathsf{E}_{X'} X_i')||_p^p \\
\leq \; & \mathsf{E}_{X,X'}||\sum_i (X_i - X_i')||_p^p \text{ Jensen} \\
= \; & \mathsf{E}_{X,X',\zeta}||\sum_i \zeta_i (X_i - X_i')||_p^p \text{ since } X_i - X_i' \text{ is a symmetric random variable} \\
\leq \; & 2^p \mathsf{E}\,||\sum_i \zeta_i X_i||_p^p + 2^p \mathsf{E}\,||\sum_i \zeta_i X_i'||_p^p = 2^{p+1} \mathsf{E}\,||\sum_i \zeta_i X_i||_p^p,
\end{aligned}$$

as claimed.

□

We would like to use the square-form theorem to prove Theorem 4.9. But this cannot be done so directly. For example, if we let X_i to be the matrix with $A^{(i)} - EA^{(i)}$ in the i'th column and 0 elsewhere, then the X_i satisfy the hypothesis of the Square Form Theorem, but unfortunately, we only get

$$\mathsf{E}\,||A - \mathsf{E}\,A||_p^p \leq \text{ some terms} + (***)\mathsf{E}\,||\sum_i X_i X_i^T||_{p/2}^{p/2},$$

which is useless since

$$||\sum_i X_i X_i^T||_{p/2}^{p/2} = ||(A - \mathsf{E}\,A)(A^T - \mathsf{E}\,A^T)||_{p/2}^{p/2} = ||A - \mathsf{E}\,A||_p^p.$$

We will actually apply the Square Form theorem with

$$X_i = (A^{(i)} - \mathsf{E}\,A^{(i)})(A^{(i)^T} - \mathsf{E}\,A^{(i)^T}) - D_i,$$

where,

$$D_i = \mathsf{E}\,\left((A^{(i)} - \mathsf{E}\,A^{(i)})(A^{(i)^T} - \mathsf{E}\,A^{(i)^T})\right).$$

Then, we have

$$\begin{aligned}
||A - \mathsf{E}\,A||_p^p &= ||\sum_i (A^{(i)} - \mathsf{E}\,A^{(i)})(A^{(i)^T} - \mathsf{E}\,A^{(i)^T})||_{p/2}^{p/2} \\
&\leq 2^{p/2}||\sum_i X_i||_{p/2}^{p/2} + 2^{p/2}||\sum_i D_i||_{p/2}^{p/2}.
\end{aligned}$$

Clearly,

$$||\sum_i D_i||_{p/2}^{p/2} \leq n^{p/2} \sum_i ||D_i||_{p/2}^{p/2} \leq n^{(p/2)+1} \sum_i \nu_i^p$$

(since D_i is a rank 1 matrix with singular value at most ν^2) which matches the second term on the right hand side of the claimed bound in the Theorem. Now, we bound $\mathsf{E}\,||\sum_i X_i||_{p/2}^{p/2}$. Let $\zeta = (\zeta_1, \zeta_2, \ldots \zeta_n)$ be independent ± 1 random variables also independent of X_i. Then by Lemma (4.14), we have (with $B^{(i)} = A^{(i)} - \mathsf{E}\,A^{(i)}$ for notational convenience)

$$\begin{aligned}
\mathsf{E}\,||\sum_i X_i||_{p/2}^{p/2} &\leq 2^{(p/2)+1}\mathsf{E}\,||\sum_i \zeta_i X_i||_{p/2}^{p/2} \\
&\leq 2^{p+1}\mathsf{E}\,||\sum_i \zeta_i B^{(i)} B^{(i)^T}||_{p/2}^{p/2} + 2^{p+1}\mathsf{E}\,||\sum_i \zeta_i D_i||_{p/2}^{p/2}.
\end{aligned}$$

The term $2^{p+1}\mathsf{E}\,||\sum_i \zeta_i D_i||_{p/2}^{p/2}$ is easy to bound as above. Now applying the square form theorem to the first term, we get

$$\begin{aligned}
&\mathsf{E}\,||\sum_i \zeta_i B^{(i)} B^{(i)^T}||_{p/2}^{p/2} \leq (cp)^{cp}\mathsf{E}\,||\sum_i B^{(i)} B^{(i)^T} B^{(i)} B^{(i)^T}||_{p/4}^{p/4} \\
= \;& (cp)^{cp}\mathsf{E}\,||\sum_i |B^{(i)}|^2 B^{(i)} B^{(i)^T}||_{p/4}^{p/4} \\
\leq \;& (cp)^{cp}\mathsf{E}\,\max_i |B^{(i)}|^{p/2} ||\sum_i B^{(i)} B^{(i)^T}||_{p/4}^{p/4} \text{ since all } B^{(i)} B^{(i)^T} \text{ are p.s.d} \\
\leq \;& (cp)^{cp} \left(\mathsf{E}\,\max_i |B^{(i)}|^p\right)^{1/2} \left(\mathsf{E}\,||\sum_i B^{(i)} B^{(i)^T}||_{p/4}^{p/2}\right)^{1/2} \text{ Jensen} \\
\leq \;& (cp)^{cp}\sqrt{n} \left(\mathsf{E}\,\max_i |B^{(i)}|^p\right)^{1/2} \left(\mathsf{E}\,||A - \mathsf{E}\,A||_p^p\right)^{1/2}
\end{aligned}$$

since

$$(\lambda_1^{p/4} + \lambda_2^{p/4} + \ldots \lambda_n^{p/4})^2 \leq n(\lambda_1^{p/2} + \lambda_2^{p/2} + \ldots \lambda_n^{p/2}).$$

Putting this all together, and letting

$$a = (\mathsf{E}\,||A - \mathsf{E}\,A||_p^p)^{1/2}, \quad b = (cp)^{cp}\sqrt{n} \left(\mathsf{E}\,\max_i |A^{(i)} - \mathsf{E}\,A^{(i)}|^p\right)^{1/2}$$

and

$$c_0 = (cp)^{cp} n^{(p/2)+1} \sum_i \nu_i^{p/2}$$

we get the following quadratic inequality on a

$$a^2 \leq ab + c_0,$$

which now implies that $a^2 \leq b^2 + 2c_0$ finishing the proof of Theorem 4.9.

4.6 Discussion

The bounds on eigenvalues of symmetric random matrices, formulated by Wigner, were proved by Füredi and Komlos [FK81] and tightened by Vu [Vu05]. Unlike the concentration based proof given here, these papers use combinatorial methods and derive sharper bounds. Spectral methods were used for planted problems by Boppana [Bop87] and Alon et al [AKS98]. Subsequently, McSherry gave a simpler algorithm for finding planted partitions [McS01], and this was refined and generalized by Coga-Oghlan [**?**]. More recently, Feige and Ron [**?**] gave a combinatorial algorithm for recovering planted cliques with similar bounds. It is an open problem to give a simple, optimal clean-up algorithm for spectral clustering.

A body of work that we have not covered here deals with limited independence, i.e., only the rows are i.i.d. but the entries of a row could be correlated. A. Dasgupta, Hopcroft, Kannan and Mitra [DHKM07] give bounds for spectral norms of such matrices based on the functional analysis work of Rudelson [Rud99] and Lust-Picard [LP86]. Specifically, the Square form Theorem and its proof are essentially in a rather terse paper by Lust-Picard [LP86]. Here we have put in all the details. The application of the Square Form Theorem to prove an analog of Theorem (4.9) is from Rudelson's work [**?**]. He deals with the case when the $A^{(i)}$ are i.i.d. and uses a different version of the Square Form Theorem (with better dependence on p) which he obtains by using some more recent methods from functional analysis. A further reference for decoupling is [**?**].

Spectral projection was also used in random topic models by Papadimitriou et al [PRTV98] and extended by Azar et al [AFKM01]. We will discuss these topics in a later chapter.

Chapter 5

Spectral Clustering

There are two main approaches to clustering using spectral methods. The first, *project-and-cluster*, is to project to the span of the top k singular vectors of the data, then apply a clustering algorithm in the k-dimensional space (e.g., k-means). The second, *partition-and-recurse*, is to use one or more singular vectors to partition the data and recurse on each part, obtaining a hiearchical clustering of the entire graph; the latter can then be pruned using a suitable objective function. In this chapter, we will see that for both methods, there are rigorous performance guarantees under suitable assumptions on the data.

5.1 Project-and-Cluster

Let the input data consisting of points in $\boldsymbol{R}^n$ be the rows of an $m \times n$ matrix A. We analyze an algorithm that will partition the data into k clusters by choosing k points called "centers", s.t. each cluster consists of the points closest to one of the k centers. We will denote a clustering by an $m \times n$ matrix C with only k distinct rows, so that the i'th row corresponds to the center assigned to the i'th row of A.

Our overall goal is to find a clustering algorithm that outputs a good clustering when one exists. We will analyze the following algorithm.

Algorithm: Project-and-Cluster

1. (project) Project data to top k-dimensional SVD subspace to get $\tilde{A}$.
2. (optimize) Use k-means or some other algorithm to find a clustering C' in $\mathbb{R}^k$ that approximately minimizes $\|\tilde{A} - C'\|_F$.
3. (merge) If for some pair of cluster centers, C'_i, C'_j, we have
$$\|C'_i - C'_j\| \leq \Delta,$$
merge clusters C'_i and C'_j into one cluster and make their center of gravity the new cluster center; repeat till no more merges can be made. (This may leave us with fewer than k clusters).

Step 2 of the algorithm asks for an approximate minimum to the k-means objective function. A constant factor solution can be found for arbitrary metrics [KMN+04]. In our setting, even running the k-means iteration after projection will converge to near-optimal solution [KK10]. Step 3 is needed only if the proper clustering might have some very small clusters; as we see in Exercise 5.1, it can be avoided by assuming no small clusters in the proper clustering.

5.1.1 Proper clusterings

To define a good clustering, we use the following parameter of a clustering C.

$$\sigma^2(C) = \frac{1}{m} \max_{v:||v||=1} ||(A-C)v||^2 = \frac{1}{m} \cdot ||A-C||_2^2$$

Roughly speaking, a good clustering C will have small $\sigma(C)$ compared to the inter-center distances.

Definition 5.1. *A clustering is called* $(1-\epsilon)$-proper *if* $\forall i, j, C_i^* \neq C_j^*$ *implies that*

$$\|C_i^* - C_j^*\| > \frac{12k^{3/2}}{\sqrt{\epsilon}}\sigma(C^*).$$

5.1.2 Performance guarantee

The algorithm is guaranteed to find a clustering that agrees with the proper clustering on almost every data point.

Theorem 5.2. *If A has a $(1-\epsilon)$-proper clustering C^*, then Project-and-Cluster with $\Delta = 10\sqrt{\frac{k}{\epsilon}}\sigma(C')$ will output a clustering C that differs from C^* in at most ϵm points. (Note that C' is the clustering produced after the second step).*

In Section 5.1.3, we give a variant of the above algorithm and analysis to avoid the merge step but assume that clusters are not too small in the proper clustering.

Exercise 5.1. *Assume that A has a $(1-\epsilon)$-proper clustering C^* where each cluster has at least $\epsilon_1 m$ points. Suppose we run Algorithm Project-and-Cluster* without *the merge step to get a clustering C. Give a bound on the number of points where C and C^* differ.*

In the proof below, we will use Lemma 4.3 from the previous chapter, which bounds the Frobenius norm error in terms of the 2-norm error.

Lemma 5.3. *Suppose C', C are any two clusterings of A such that*

$$\|\tilde{A} - C'\|_F^2 \leq 2\|\tilde{A} - C\|_F^2.$$

Then $\sigma(C') \leq 4\sqrt{k}\sigma(C)$.

Proof.

$$\begin{aligned}
\sqrt{m}\sigma(C') &= \|A - C'\|_2 \\
&\leq \|A - \tilde{A}_k\|_2 + \|\tilde{A}_k - C'\|_2 \\
&\leq \sqrt{m}\sigma(C) + \sqrt{2}\|\tilde{A}_k - C\|_F \text{ (factor-2 } k\text{-means approximation)} \\
&\leq \sqrt{m}\sigma(C) + \sqrt{2}\sqrt{5k}\|A - C\|_2 \text{ (using Lemma 4.3)} \\
&\leq \sqrt{m}\sigma(C) + \sqrt{10k}\sqrt{m}\sigma(C) \\
&\leq 4\sqrt{k}(\sqrt{m}\sigma(C)).
\end{aligned}$$

□

The next lemma sounds a bit surprising since it holds for *any* clustering C, but the bound depends on $\sigma(C)$.

Lemma 5.4. *Let C be any clustering and C' be the clustering found by Algorithm Project-and-Cluster after the second step. For all but ϵm points, we have*

$$\|C_i' - C_i\| < 6\sqrt{\frac{k}{\epsilon}}\sigma(C).$$

Proof. Let B denote points that do not satisfy the conclusion of the lemma:

$$B : \{i : \|C_i' - C_i\|^2 \geq \frac{36k}{\epsilon}\sigma^2(C)\}.$$

We can lower bound the squared distance of these points from their assigned centers. We use the elementary inequality: $(a-b)^2 \geq (a^2/2) - b^2$.

$$\begin{aligned}
\sum_{i \in B} \|\tilde{A}_i - C_i'\|^2 &= \sum_B \|\tilde{A}_i - C_i + C_i - C'\|^2 \\
&\geq \frac{1}{2}\sum_B \|C_i - C_i'\|^2 - \sum_B \|\tilde{A}_i - C_i\|^2 \\
&\geq \frac{36k|B|}{2\epsilon}\sigma^2(C') - \|\tilde{A} - C\|_F^2
\end{aligned}$$

We can also bound this sum of squared distances from above3 using the fact that the algorithm produces a 2-approximation to the k-means objective in the projected space, and therefore to any k-means solution:

$$\sum_{i \in B} \|\tilde{A}_i - C'_i\|^2 \quad \leq \quad \|\bar{A} - C'\|_F^2 \leq 2\|\tilde{A} - C\|_F^2$$

Combining these bounds with the assumption on C, we have

$$\begin{aligned} |B| \cdot \frac{36k}{2\epsilon}\sigma^2(C') &\leq 3\|\tilde{A} - C\|_F^2 \\ &\leq 15k\|A - C\|_2^2 \text{ (using Lemma 4.3)} \\ &= 15km\sigma^2(C). \end{aligned}$$

From this it follows that $|B| \leq (30/36)\epsilon m < \epsilon m$. □

Proof of Theorem 5.2. Let C' be the clustering found by the Algorithm Project-and-Cluster after the second step. Let $\Delta = 12\sqrt{\frac{k}{\epsilon}}\sigma(C^*)$ and define

$$B = \{i : \|C'_i - C^*_i\| > \Delta/2\}.$$

By Lemma 5.4, $|B| < \epsilon m$.

For any two points $i, j \notin B$ and from the same cluster of C^* (i.e., $C^*_i = C^*_j$), we have

$$\begin{aligned} \|C'_i - C'_j\| &\leq \|C'_i - C^*_i\| + \|C'_j - C^*_j\| + \|C^*_i - C^*_j\| \\ &\leq \Delta. \end{aligned}$$

Therefore i, j will be in the same final cluster output by the algorithm as well.

For $i, j \notin B$, from different clusters of C^*, using Lemma 5.3 applied to C' and C^*,

$$\|C^*_i - C^*_j\| \geq \frac{12k^{3/2}}{\sqrt{\epsilon}}\sigma(C^*) \geq k(12\sqrt{\frac{k}{\epsilon}}\sigma(C^*)) = k\Delta.$$

Therefore,

$$\begin{aligned} \|C'_i - C'_j\| &> \|C^*_i - C^*_j\| - \|C'_i - C^*_i\| - \|C'_j - C^*_j| \geq k\Delta - \Delta \\ &\geq (k-1)\Delta. \end{aligned}$$

after one merge. Since there are at most $k - 1$ merges, this guarantees that i and j will be in separate clusters at the end of the algorithm. Thus, the final clustering C output by the algorithm and any proper clustering C^* agree on all but ϵm points. □

5.1.3 Project-and-Cluster Assuming No Small Clusters

We will analyze the following algorithm.

Algorithm: Spectral-Clustering

1. (project) Project data to top k-dimensional SVD subspace to get $\tilde{A}$.
2. (optimize) Pick a random row of $\tilde{A}$. Include all points within distance $\frac{6k}{\epsilon}\sigma(C)$ in one cluster. Repeat k times.

The algorithm is guaranteed to find a clustering that agrees with the proper clustering on almost every data point.

Theorem 5.5. *For $\epsilon > 0$, if A has a clustering C^* such that $\|C_i^* - C_j^*\| > \frac{15k}{\epsilon}\sigma(C^*)$ and each cluster has at least ϵnpoints, then with probability at least $1-\epsilon$, Spectral-Clustering will output a clustering C that differs from C^* in at most $\epsilon^2 m$ points.*

Lemma 5.6. *Suppose A, B are $m \times n$ matrices with rank$(B) = k$. If $\tilde{A}$ is the best rank k approximation to A, then*

$$\|\tilde{A} - B\|_F^2 \leq 8k\|A - B\|^2.$$

Proof. Since $\tilde{A} - B$ has rank at most $2k$, and $\hat{A}$ is the best rank k approximation of A, we have,

$$\begin{aligned} \|\tilde{A} - B\|_F^2 &\leq 2k\|\tilde{A} - B\|_2^2 \\ &\leq 2k\left(\|\tilde{A} - A\|_2 + \|A - B\|_2\right)^2 \\ &\leq 8k\|A - B\|_2^2. \end{aligned}$$

□

Proof of Theorem 5.5. Suppose v_i is the i'th row of $\tilde{A}$. We claim that most v_i are within distance $\frac{3k}{\epsilon}\sigma(C)$ of their center c_i.

Let $B = \{i : \|v_i - c_i\| > \frac{3k}{\epsilon}\sigma(C)\}$. Then

$$8k\sigma(C)^2 m \geq \|\tilde{A} - C\|_F^2 \geq |B|\frac{9k^2}{\epsilon^2}\sigma(C)^2.$$

Thus $|B| < \frac{\epsilon^2}{k}m$.

For i, j in the same cluster and $i, j \notin B$,

$$\|v_i - v_j\| \leq \frac{6k}{\epsilon}\sigma(C).$$

For i, j in different clusters and $i, j \notin B$,

$$\|v_i - v_j\| > \frac{15k}{\epsilon}\sigma(C) - \frac{6k}{\epsilon}\sigma(C) = \frac{9k}{\epsilon}\sigma(C).$$

Hence if we pick point not in B as the seed, all k times, then all points not in B will be correctly classified. The success probability is

$$\mathbf{Pr}(k \text{ random points} \notin B) \geq \left(1 - \frac{\epsilon^2}{k}\right)\left(1 - \frac{\epsilon - \epsilon^2}{k}\right)^{k-1} \geq 1 - \frac{\epsilon}{k}k = 1 - \epsilon.$$

□

5.2 Partition-and-Recurse

Next we study a spectral algorithm for recursively partitioning a graph. The key algorithmic ingredient is a procedure to find an approximately minimum conductance cut. This cutting procedure is used recursively to obtain a clustering algorithm. The analysis is based on a natural bicriteria measure for assessing the quality of a clustering and makes no probabilistic assumptions on the input data.

We begin with an important definition that plays a key role both in graph partitioning and in the analysis of Markov chains. Given a graph $G = (V, E)$, with nonnegative edge weights a_{ij}, for a subset of vertices S, we let $a(S)$ denote the total weight of edges incident to vertices in S. Then the conductance of a subset S is

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} a_{ij}}{\min\{a(S), a(V \setminus S)\}},$$

and the conductance of the graph is

$$\phi = \min_{S \subset V} \phi(S).$$

To see this in the context of Markov chains, let A be the nonnegative weighted adjacency matrix and B be obtained by normalizing each row to have to sum equal to 1. Then it follows that the largest eigenvalue of B is 1 and (assuming ergodicity), the stationary distribution π is simply proportional to $a(i)$, i.e.,

$$B^T \pi = \pi.$$

Now for any subset of vertices $S \subset V$,

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} \pi_i b_{ij}}{\min\{\pi(S), \pi(V \setminus S)\}}$$

and ϕ controls the rate of convergence of the Markov chain defined by the transition matrix B to its stationary distribution π.

5.2.1 Approximate minimum conductance cut

The following simple algorithm takes a weighted graph (or weighted adjacency matrix) as input and outputs a cut of the graph.

Algorithm: Approximate-Cut

1. Normalize the adjancency matrix so each row sum is 1.
2. Find the second largest eigenvector of this matrix.
3. Order the vertices according their components in this vector.
4. Find the minimum conductance cut among cuts given by this ordering.

The following theorem bounds the conductance of the cut found by this heuristic with respect to the minimum conductance. This theorem plays an important role in the analysis of Markov chains, where conductance is often easier to estimate than the desired quantity, the spectral gap. The latter determines the mixing rate of the Markov chain. Later in this chapter, we will use this cutting procedure as a tool to find a clustering.

Theorem 5.7. *Suppose B is a $N \times N$ matrix with non-negative entries with each row sum equal to 1 and suppose there are positive real numbers $\pi_1, \pi_2, \ldots \pi_N$ summing to 1 such that $\pi_i b_{ij} = \pi_j b_{ji}$ for all i, j. If v is the right eigenvector of B corresponding to the second largest eigenvalue λ_2, and $i_1, i_2, \ldots i_N$ is an ordering of $1, 2, \ldots N$ so that $v_{i_1} \geq v_{i_2} \ldots \geq v_{i_N}$, then*

$$2 \min_{S \subset V} \phi(S) \geq 1 - \lambda_2 \geq \frac{1}{2} \left(\min_{l, 1 \leq l \leq N} \phi(\{1, 2, \ldots, l\}) \right)^2$$

We note here that the leftmost term above is just the conductance of the graph with weights b_{ij}, while the rightmost term is the square of the minimum conductance of cuts along the ordering given by the second eigenvector of the of the normalized adjacency matrix. Since the latter is trivially at least as large as the square of the overall minimum conductance, we get

$$2\text{min conductance} \geq 1 - \lambda_2 \geq \frac{1}{2} \left(\text{min conductance}\right)^2.$$

Proof (of Theorem 5.7). We first evaluate the second eigenvalue. Towards this end, let $D^2 = \text{diag}(\pi)$. Then, from the time-reversibility property of B, we have $D^2 B = B^T D^2$. Hence $Q = DBD^{-1}$ is symmetric. The eigenvalues of B and Q are the same, with their largest eigenvalue equal to 1. In addition, $\pi^T D^{-1} Q = \pi^T D^{-1}$ and therefore $\pi^T D^{-1}$ is the left eigenvector of Q corresponding to the eigenvalue 1. So we have,

$$\lambda_2 = \max_{\pi^T D^{-1} x = 0} \frac{x^T DBD^{-1} x}{x^T x}$$

Thus, substituting $y = D^{-1}x$, we obtain

$$1 - \lambda_2 = \min_{\pi^T D^{-1} x = 0} \frac{x^T D(I - B)D^{-1} x}{x^T x} = \min_{\pi^T y = 0} \frac{y^T D^2 (I - B) y}{y^T D^2 y}$$

The numerator can be rewritten:

$$\begin{aligned} y^T D^2(I-B)y &= -\sum_{i\neq j} y_i y_j \pi_i b_{ij} + \sum_i \pi_i(1-b_{ii})y_i^2 \\ &= -\sum_{i\neq j} y_i y_j \pi_i b_{ij} + \sum_{i\neq j} \pi_i b_{ij} \frac{y_i^2+y_j^2}{2} \\ &= \sum_{i<j} \pi_i b_{ij}(y_i-y_j)^2 \end{aligned}$$

Denote this final term by $\mathcal{E}(y,y)$. Then

$$1-\lambda_2 = \min_{\pi^T y=0} \frac{\mathcal{E}(y,y)}{\sum_i \pi_i y_i^2}$$

To prove the first inequality of the theorem, let $(S,\bar{S})$ be the cut with the minimum conductance. Define a vector w as follows

$$w_i = \begin{cases} \sqrt{\frac{\pi(\bar{S})}{\pi(S)}} & \text{if } i \in S \\ -\sqrt{\frac{\pi(S)}{\pi(\bar{S})}} & \text{if } i \in \bar{S} \end{cases}$$

It is then easy to check that $\sum_i \pi_i w_i = 0$ and that

$$\phi(S) \geq \frac{\mathcal{E}(w,w)}{\sum_i \pi_i w_i^2} \geq \frac{1-\lambda_2}{2}.$$

Hence we obtain the desired lower bound on the conductance.

We will now prove the second inequality. Suppose that the minimum above is attained when y is equal to v. Then Dv is the eigenvector of Q corresponding to the eigenvalue λ_2 and, v is the right eigenvector of B corresponding to λ_2. Our ordering is then with respect to v in accordance with the statement of the theorem. Assume that, for simplicity of notation, the indices are reordered (i.e. the rows and corresponding columns of B and D are reordered) so that

$$v_1 \geq v_2 \geq \cdots \geq v_N.$$

Now define r to satisfy

$$\pi_1 + \pi_2 + \cdots + \pi_{r-1} \leq \frac{1}{2} < \pi_1 + \pi_2 + \cdots + \pi_r,$$

and let $z_i = v_i - v_r$ for $i = 1, \ldots, n$. Then

$$z_1 \geq z_2 \geq \cdots \geq z_r = 0 \geq z_{r+1} \geq \cdots \geq z_n,$$

and

$$\begin{aligned}\frac{\mathcal{E}(v,v)}{\sum_i \pi_i v_i^2} &= \frac{\mathcal{E}(z,z)}{-v_r^2 + \sum_i \pi_i z_i^2} \\ &\geq \frac{\mathcal{E}(z,z)}{\sum_i \pi_i z_i^2} \\ &= \frac{\left(\sum_{i<j} \pi_i b_{ij}(z_i - z_j)^2\right)\left(\sum_{i<j} \pi_i b_{ij}(|z_i| + |z_j|)^2\right)}{\left(\sum_i \pi_i z_i^2\right)\left(\sum_{i<j} \pi_i b_{ij}(|z_i| + |z_j|)^2\right)}\end{aligned}$$

Consider the numerator of this final term. By Cauchy-Schwartz

$$\begin{aligned}\left(\sum_{i<j} \pi_i b_{ij}(z_i - z_j)^2\right)\left(\sum_{i<j} \pi_i b_{ij}(|z_i| + |z_j|)^2\right) &\geq \left(\sum_{i<j} \pi_i b_{ij}|z_i - z_j|(|z_i| + |z_j|)\right)^2 \\ &\geq \left(\sum_{i<j} \pi_i b_{ij} \sum_{k=i}^{j-1} |z_{k+1}^2 - z_k^2|\right)^2 \quad (5.1)\end{aligned}$$

Here the second inequality follows from the fact that if $i < j$ then

$$|z_i - z_j|(|z_i| + |z_j|) \geq \sum_{k=i}^{j-1} |z_{k+1}^2 - z_k^2|.$$

This follows from the following observations:

a. If z_i and z_j have the same sign (i.e. $r \notin \{i, i+1, \ldots, j\}$) then

$$|z_i - z_j|(|z_i| + |z_j|) = |z_i^2 - z_j^2|.$$

b. Otherwise, if z_i and z_j have different signs then

$$|z_i - z_j|(|z_i| + |z_j|) = (|z_i| + |z_j|)^2 > z_i^2 + z_j^2.$$

Also,

$$\sum_{i<j} \pi_i b_{ij}(|z_i| + |z_j|)^2 \leq 2\sum_{i<j} \pi_i b_{ij}(z_i^2 + z_j^2) \leq 2\sum_i \pi_i z_i^2$$

As a result we have,

$$\begin{aligned}\frac{\mathcal{E}(v,v)}{\sum_i \pi_i v_i^2} &\geq \frac{\left(\sum_{i<j} \pi_i b_{ij}(z_i - z_j)^2\right)\left(\sum_{i<j} \pi_i b_{ij}(|z_i| + |z_j|)^2\right)}{\left(\sum_i \pi_i z_i^2\right)\left(\sum_{i<j} \pi_i b_{ij}(|z_i| + |z_j|)^2\right)} \\ &\geq \frac{\left(\sum_{i<j} \pi_i b_{ij} \sum_{k=i}^{j-1} |z_{k+1}^2 - z_k^2|\right)^2}{2\left(\sum_i \pi_i z_i^2\right)^2}\end{aligned}$$

Set $S_k = \{1, 2, \ldots, k\}$, $C_k = \{(i,j) : i \leq k < j\}$ and

$$\hat{\alpha} = \min_{k, 1 \leq k \leq N} \frac{\sum_{(i,j) \in C_k} \pi_i b_{ij}}{\min(\sum_{i: i \leq k} \pi_i, \sum_{i: i > k} \pi_i)}$$

Since $z_r = 0$, we obtain

$$\begin{aligned}
\sum_{i<j} \pi_i b_{ij} \sum_{k=i}^{j-1} |z_{k+1}^2 - z_k^2| &= \sum_{k=1}^{N-1} |z_{k+1}^2 - z_k^2| \sum_{(i,j) \in C_k} \pi_i b_{ij} \\
&\geq \hat{\alpha} \left(\sum_{k=1}^{r-1} (z_k^2 - z_{k+1}^2) \pi(S_k) + \sum_{k=r}^{N-1} (z_{k+1}^2 - z_k^2)(1 - \pi(S_k)) \right) \\
&= \hat{\alpha} \left(\sum_{k=1}^{N-1} (z_k^2 - z_{k+1}^2) \pi(S_k) + (z_N^2 - z_r^2) \right) \\
&= \hat{\alpha} \sum_{k=1}^{N} \pi_k z_k^2.
\end{aligned}$$

Consequently, if $\pi^T y = 0$ then

$$1 - \lambda_2 = \frac{\mathcal{E}(v,v)}{\sum_i \pi_i v_i^2} \geq \frac{\hat{\alpha}^2}{2}.$$

□

5.2.2 Two criteria to measure the quality of a clustering

The measure of the quality of a clustering we will use here is based on expansion-like properties of the underlying pairwise similarity graph. The quality of a clustering is given by two parameters: α, the minimum conductance of the clusters, and ϵ, the ratio of the weight of inter-cluster edges to the total weight of all edges. Roughly speaking, a good clustering achieves high α and low ϵ. Note that the conductance provides a measure of the quality of an individual cluster (and thus of the overall clustering) while the weight of the inter-cluster edges provides a measure of the cost of the clustering. Hence, imposing a lower bound, α, on the quality of each individual cluster we seek to minimize the cost, ϵ, of the clustering; or conversely, imposing an upper bound on the cost of the clustering we strive to maximize its quality. For a detailed motivation of this bicriteria measure we refer the reader to the introduction of [KVV04].

Definition 5.8. *We call a partition $\{C_1, C_2, \ldots, C_l\}$ of V an (α, ϵ)-clustering if:*
1. The conductance of each C_i is at least α.
2. The total weight of inter-cluster edges is at most an ϵ fraction of the total edge weight.

Associated with this bicriteria measure is the following optimization problem: (P1) Given α, find an (α, ϵ)-clustering that minimizes ϵ (alternatively, we have (P2) Given ϵ, find an (α, ϵ)-clustering that maximizes α). We note that the number of clusters is not restricted.

5.2.3 Approximation Algorithms

Problem (P1) is NP-hard. To see this, consider maximizing α with ϵ set to zero. This problem is equivalent to finding the conductance of a given graph which is well known to be NP-hard [GJ79]. We consider the following heuristic approach.

Algorithm: Recursive-Cluster

1. Find a cut that approximates the minimum conductance cut in G.
2. If the conductance of the cut obtained is below a preset threshold, recurse on the pieces induced by the cut.

The idea behind our algorithm is simple. Given G, find a cut $(S, \bar{S})$ of minimum conductance. Then recurse on the subgraphs induced by S and $\bar{S}$. Finding a cut of minimum conductance is hard, and hence we need to use an approximately minimum cut. There are two well-known approximations for the minimum conductance cut, one is based on a semidefinite programming relaxation (and precurson on a linear programming relaxation) and the other is derived from the second eigenvector of the graph. Before we discuss these approximations, we present a general theorem that captures both for the purpose of analyzing the clustering heuristic.

Let $\mathcal{A}$ be an approximation algorithm that produces a cut of conductance at most Kx^{ν} if the minimum conductance is x, where K is independent of x (K could be a function of n, for example) and ν is a fixed constant between between 0 and 1. The following theorem provides a guarantee for the approximate-cluster algorithm using $\mathcal{A}$ as a subroutine.

Theorem 5.9. *If G has an (α, ϵ)-clustering, then the recursive-cluster algorithm, using approximation algorithm $\mathcal{A}$ as a subroutine, will find a clustering of quality*

$$\left(\left(\frac{\alpha}{6K \log \frac{n}{\epsilon}} \right)^{1/\nu}, (12K+2)\epsilon^{\nu} \log \frac{n}{\epsilon} \right).$$

Proof. Let the cuts produced by the algorithm be $(S_1, T_1), (S_2, T_2), \ldots$, where we adopt the convention that S_j is the "smaller" side (i.e., $a(S_j) \leq a(T_j)$). Let $C_1, C_2, \ldots C_l$ be an (α, ϵ)-clustering. We use the termination condition of $\alpha^* = \frac{\alpha}{6 \log \frac{n}{\epsilon}}$. We will assume that we apply the recursive step in the algorithm

only if the conductance of a given piece as detected by the heuristic for the minimum conductance cut is less than α^*. In addition, purely for the sake of analysis we consider a slightly modified algorithm. If at any point we have a cluster C_t with the property that $a(C_t) < \frac{\epsilon}{n} a(V)$ then we split C_t into singletons. The conductance of singletons is defined to be 1. Then, upon termination, each cluster has conductance at least

$$\left(\frac{\alpha^*}{K}\right)^{1/\nu} = \left(\frac{\alpha}{6K \log \frac{n}{\epsilon}}\right)^{1/\nu}$$

Thus it remains to bound the weight of the inter-cluster edges. Observe that $a(V)$ is twice the total edge weight in the graph, and so $W = \frac{\epsilon}{2} a(V)$ is the weight of the inter-cluster edges in this optimal solution.

Now we divide the cuts into two groups. The first group, H, consists of cuts with "high" conductance within clusters. The second group consists of the remaining cuts. We will use the notation $w(S_j, T_j) = \sum_{u \in S_j, v \in T_j} a_{uv}$. In addition, we denote by $w_{\mathrm{I}}(S_j, T_j)$ the sum of the weights of the intra-cluster edges of the cut (S_j, T_j), i.e., $w_{\mathrm{I}}(S_j, T_j) = \sum_{i=1}^{l} w(S_j \cap C_i, T_j \cap C_i)$. We then set

$$H = \left\{ j : w_{\mathrm{I}}(S_j, T_j) \geq 2\alpha^* \sum_{i=1}^{l} \min(a(S_j \cap C_i), a(T_j \cap C_i)) \right\}$$

We now bound the cost of the high conductance group. For all $j \in H$, we have,

$$\alpha^* a(S_j) \quad \geq \quad w(S_j, T_j) \geq w_{\mathrm{I}}(S_j, T_j) \geq 2\alpha^* \sum_i \min(a(S_j \cap C_i), a(T_j \cap C_i))$$

Consequently we observe that

$$\sum_i \min(a(S_j \cap C_i), a(T_j \cap C_i)) \leq \frac{1}{2} a(S_j)$$

From the algorithm's cuts, $\{(S_j, T_j)\}$, and the optimal clustering, $\{C_i\}$, we define a new clustering via a set of cuts $\{(S'_j, T'_j)\}$ as follows. For each $j \in H$, we define a cluster-avoiding cut (S'_j, T'_j) in $S_j \cup T_j$ in the following manner. For each $i, 1 \leq i \leq l$, if $a(S_j \cap C_i) \geq a(T_j \cap C_i)$, then place all of $(S_j \cup T_j) \cap C_i$ into S'_j. If $a(S_j \cap C_i) < a(T_j \cap C_i)$, then place all of $(S_j \cup T_j) \cap C_i$ into T'_j.

Notice that, since $|a(S_j) - a(S'_j)| \leq \frac{1}{2} a(S_j)$, we have that $\min(a(S'_j), a(T'_j)) \geq \frac{1}{2} a(S_j)$. Now we will use the approximation guarantee for the cut procedure to get an upper bound on $w(S_j, T_j)$ in terms of $w(S'_j, T'_j)$.

$$\begin{aligned} \frac{w(S_j, T_j)}{a(S_j)} &\leq K \left(\frac{w(S'_j, T'_j)}{\min\{a(S'_j), a(T'_j)\}} \right)^{\nu} \\ &\leq K \left(\frac{2w(S'_j, T'_j)}{a(S_j)} \right)^{\nu} \end{aligned}$$

Hence we have bounded the overall cost of the high conductance cuts with respect to the cost of the cluster-avoiding cuts. We now bound the cost of these cluster-avoiding cuts. Let $P(S)$ denote the set of inter-cluster edges incident at a vertex in S, for any subset S of V. Also, for a set of edges F, let $w(F)$ denote the sum of their weights. Then, $w(S'_j, T'_j) \leq w(P(S'_j))$, since every edge in (S'_j, T'_j) is an inter-cluster edge. So we have,

$$w(S_j, T_j) \quad \leq \quad K\big(2w(P(S'_j))\big)^{\nu} a(S_j)^{1-\nu} \tag{5.2}$$

Next we prove the following claim.

Claim 1. For each vertex $u \in V$, there are at most $\log \frac{n}{\epsilon}$ values of j such that u belongs to S_j. Further, there are at most $2 \log \frac{n}{\epsilon}$ values of j such that u belongs to S'_j.

To prove the claim, fix a vertex $u \in V$. Let

$$I_u = \{j : u \in S_j\} \qquad\qquad J_u = \{j : u \in S'_j \setminus S_j\}$$

Clearly if $u \in S_j \cap S_k$ (with $k > j$), then (S_k, T_k) must be a partition of S_j or a subset of S_j. Now we have, $a(S_k) \leq \frac{1}{2} a(S_k \cup T_k) \leq \frac{1}{2} a(S_j)$. So $a(S_j)$ reduces by a factor of 2 or greater between two successive times u belongs to S_j. The maximum value of $a(S_j)$ is at most $a(V)$ and the minimum value is at least $\frac{\epsilon}{n} a(V)$, so the first statement of the claim follows.

Now suppose $j, k \in J_u; j < k$. Suppose also $u \in C_i$. Then $u \in T_j \cap C_i$. Also, later, T_j (or a subset of T_j) is partitioned into (S_k, T_k) and, since $u \in S'_k \setminus S_k$, we have $a(T_k \cap C_i) \leq a(S_k \cap C_i)$. Thus $a(T_k \cap C_i) \leq \frac{1}{2} a(S_k \cup T_k) \leq \frac{1}{2} a(T_j \cap C_i)$. Thus $a(T_j \cap C_i)$ halves between two successive times that $j \in J_u$. So, $|J_u| \leq \log \frac{n}{\epsilon}$. This proves the second statement in the claim (since $u \in S'_j$ implies that $u \in S_j$ or $u \in S'_j \setminus S_j$).

Using this claim, we can bound the overall cost of the group of cuts with high conductance within clusters with respect to the cost of the optimal clustering as follows:

$$\begin{aligned}
\sum_{j \in H} w(S_j, T_j) \quad &\leq \quad \sum_{\text{all } j} K\big(2w(P(S'_j))\big)^{\nu} a(S_j)^{1-\nu} \\
&\leq \quad K\Big(2 \sum_{\text{all } j} w(P(S'_j))\Big)^{\nu} \Big(\sum_j a(S_j)\Big)^{1-\nu} \\
&\leq \quad K\left(2\epsilon \log \frac{n}{\epsilon}\, a(V)\right)^{\nu} \left(2 \log \frac{n}{\epsilon}\, a(V)\right)^{1-\nu} \\
&\leq \quad 2K\epsilon^{\nu} \log \frac{n}{\epsilon}\, a(V)
\end{aligned} \tag{5.3}$$

Here we used Hölder's inequality: for real sequences $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$, and any $p, q \geq 1$ with $(1/p) + (1/q) = 1$, we have

$$\sum_{i=1}^{n} a_i b_i \leq \left(\sum_{i=1}^{n} a_i^p\right)^{\frac{1}{p}} \left(\sum_{i=1}^{n} b_i^q\right)^{\frac{1}{q}}.$$

Next we deal with the group of cuts with low conductance within clusters i.e., those j not in H. First, suppose that all the cuts together induce a partition of C_i into $P_1^i, P_2^i, \ldots P_{r_i}^i$. Every edge between two vertices in C_i which belong to different sets of the partition must be cut by some cut (S_j, T_j) and, conversely, every edge of every cut $(S_j \cap C_i, T_j \cap C_i)$ must have its two end points in different sets of the partition. So, given that C_i has conductance α, we obtain

$$\sum_{\text{all } j} w_{\mathrm{I}}(S_j \cap C_i, T_j \cap C_i) = \frac{1}{2} \sum_{s=1}^{r_i} w(P_s^i, C_i \setminus P_s^i) \geq \frac{1}{2} \alpha \sum_s \min(a(P_s^i), a(C_i \setminus P_s^i))$$

For each vertex $u \in C_i$ there can be at most $\log \frac{n}{\epsilon}$ values of j such that u belongs to the smaller (according to $a(\cdot)$) of the two sets $S_j \cap C_i$ and $T_j \cap C_i$. So, we have that

$$\sum_{s=1}^{r_i} \min(a(P_s^i), a(C_i \setminus P_s^i)) \quad \geq \quad \frac{1}{\log \frac{n}{\epsilon}} \sum_j \min(a(S_j \cap C_i), a(T_j \cap C_i))$$

Thus,

$$\sum_{\text{all } j} w_{\mathrm{I}}(S_j, T_j) \geq \frac{\alpha}{2 \log \frac{n}{\epsilon}} \sum_{i=1}^{l} \sum_j \min(a(S_j \cap C_i), a(T_j \cap C_i))$$

Therefore, from the definition of H, we have

$$\sum_{j \notin H} w_{\mathrm{I}}(S_j, T_j) \leq 2\alpha^* \sum_{\text{all } j} \sum_{i=1}^{l} \min(a(S_j \cap C_i), a(T_j \cap C_i)) \leq \frac{2}{3} \sum_{\text{all } j} w_{\mathrm{I}}(S_j, T_j)$$

Thus, we are able to bound the intra-cluster cost of the low conductance group of cuts in terms of the intra-cluster cost of the high conductance group. Applying (5.3) then gives

$$\sum_{j \notin H} w_{\mathrm{I}}(S_j, T_j) \leq 2 \sum_{j \in H} w_{\mathrm{I}}(S_j, T_j) \leq 4K\epsilon^{\nu} \log \frac{n}{\epsilon} a(V) \tag{5.4}$$

In addition, since each inter-cluster edge belongs to at most one cut S_j, T_j, we have that

$$\sum_{j \notin H} (w(S_j, T_j) - w_{\mathrm{I}}(S_j, T_j)) \quad \leq \quad \frac{\epsilon}{2} a(V) \tag{5.5}$$

We then sum up (5.3), (5.4) and (5.5). To get the total cost we note that splitting up all the V_t with $a(V_t) \leq \frac{\epsilon}{n} a(V)$ into singletons costs us at most $\frac{\epsilon}{2} a(V)$ on the whole. Substituting $a(V)$ as twice the total sum of edge weights gives the bound on the cost of inter-cluster edge weights. This completes the proof of Theorem 5.9. □

The Leighton-Rao algorithm for approximating the conductance finds a cut of conductance at most $2 \log n$ times the minimum [LR99]. In our terminology, it is an approximation algorithm with $K = 2 \log n$ and $\nu = 1$. Applying theorem 5.9 leads to the following guarantee.

Corollary 5.10. *If the input has an (α, ϵ)-clustering, then, using the Leighton-Rao method for approximating cuts, the recursive-cluster algorithm finds an*

$$\left(\frac{\alpha}{12 \log n \log \frac{n}{\epsilon}}, 26\epsilon \log n \log \frac{n}{\epsilon}\right) \text{-clustering.}$$

We now assess the running time of the algorithm using this heuristic. The fastest implementation for this heuristic runs in $\tilde{O}(n^2)$ time (where the $\tilde{O}$ notation suppresses factors of $\log n$). Since the algorithm makes less than n cuts, the total running time is $\tilde{O}(n^3)$. This might be slow for some real-world applications. We discuss a potentially more practical algorithm in the next section. We conclude this section with the guarantee obtained using Arora et al.'s improved approximation [ARV04] of $O(\sqrt{\log n})$.

Corollary 5.11. *If the input to the recursive-cluster algorithm has an (α, ϵ)-clustering, then using the ARV method for approximating cuts, the algorithm finds an*

$$\left(\frac{\alpha}{C\sqrt{\log n} \log \frac{n}{\epsilon}}, C\epsilon \sqrt{\log n} \log \frac{n}{\epsilon}\right) \text{-clustering.}$$

where C is a fixed constant.

5.2.4 Worst-case guarantees for spectral clustering

In this section, we describe and analyze a recursive variant of the spectral algorithm. This algorithm, outlined below, has been used in computer vision, medical informatics, web search, spam detection etc.. We note that the algorithm is a special case of the recursive-cluster algorithm described in the previous section; here we use a spectral heuristic to approximate the minimum conductance cut. We assume the input is a weighted adjacency matrix A.

Algorithm: Recursive-Spectral

1. Normalize A to have unit row sums and find its second right eigenvector v.
2. Find the best ratio cut along the ordering given by v.
3. If the value of the cut is below a chosen threshold, then recurse on the pieces induced by the cut.

Thus, we find a clustering by repeatedly solving a one-dimensional clustering problem. Since the latter is easy to solve, the algorithm is efficient. The fact that it also has worst-case quality guarantees is less obvious.

We now elaborate upon the basic description of this variant of the spectral algorithm. Initially, we normalize our matrix A by scaling the rows so that the

row sums are all equal to one. At any later stage in the algorithm we have a partition $\{C_1, C_2, \ldots, C_s\}$. For each C_t, we consider the $|C_t| \times |C_t|$ submatrix B of A restricted to C_t. We normalize B by setting b_{ii} to $1 - \sum_{j \in C_t, j \neq i} b_{ij}$. As a result, B is also non-negative with row sums equal to one.

Observe that upon normalization of the matrix, our conductance measure corresponds to the familiar Markov Chain conductance measure i.e.

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} a_{ij}}{\min(a(S), a(\bar{S}))} = \frac{\sum_{i \in S, j \notin S} \pi_i b_{ij}}{\min(\pi(S), \pi(\bar{S}))}$$

where π is the stationary distribution of the Markov Chain.

We then find the second eigenvector of B. This is the right eigenvector v corresponding to the second largest eigenvalue λ_2, i.e. $Bv = \lambda_2 v$. Then order the elements (rows) of C_t decreasingly with respect to their component in the direction of v. Given this ordering, say $\{u_1, u_2, \ldots u_r\}$, find the minimum *ratio cut* in C_t. This is the cut that minimises $\phi(\{u_1, u_2, \ldots u_j\}, C_t)$ for some j, $1 \leq j \leq r-1$. We then recurse on the pieces $\{u_1, \ldots, u_j\}$ and $C_t \setminus \{u_1, \ldots, u_j\}$.

We combine Theorem 5.7 with Theorem 5.9 to get a worst-case guarantee for Algorithm Recursive-Spectral. In the terminology of Theorem 5.9, Theorem 5.7 says that the spectral heuristic for minimum conductance is an approximation algorithm with $K = \sqrt{2}$ and $\nu = 1/2$.

Corollary 5.12. *If the input has an (α, ϵ)-clustering, then, using the spectral heuristic, the approximate-cluster algorithm finds an*

$$\left(\frac{\alpha^2}{72 \log^2 \frac{n}{\epsilon}}, 20\sqrt{\epsilon} \log \frac{n}{\epsilon} \right) \text{-clustering.} \quad \square$$

5.3 Discussion

The first part of this chapter is drawn partly from the work of Kumar and Kannan [KK10], who analyzed the k-means algorithm after spectral projection. Their analysis was improved significantly by Awasthi and Or [AS12]. There is much work on the k-means algorithm and objective, and it is widely used in practice. However, at this time, our understanding of when it can be effectively applied is not complete.

The second part of this chapter is based on Kannan et al. [KVV04] and earlier work by Jerrum and Sinclair [SJ89]. Theorem 5.7 was essentially proved by Sinclair and Jerrum (in their proof of Lemma 3.3 in [SJ89], although not mentioned in the statement of the lemma). Cheng et al. [CKVW06] give an efficient implementation of recursive-spectral that maintains sparsity, and has been used effectively on large data sets from diverse applications.

Spectral partitioning has also been shown to have good guarantees for some special classes of graphs. Notably, Spielman and Teng [ST07] proved that a variant of spectral partitioning produces small separators for bounded-degree planar graphs, which often come up in practical applications of spectral cuts.

The key contribution of their work was an upper bound on the second smallest eigenvalue of the Laplacian of a planar graph. This work was subsequently generalized to graphs of bounded genus [Kel06].

Chapter 6

Combinatorial Optimization via Low-Rank Approximation

Part II

Algorithms

Chapter 7

Power Iteration

Let $T \in \otimes^3 \boldsymbol{R}^d$ have an orthogonal decomposition $T = \sum_{i=1}^k \lambda_i v_i \otimes v_i \otimes v_i$ with $\|v_i\| = 1$ for all $i \in [k]$. For a random choice of $\theta \in \boldsymbol{R}^d$, for example from a spherical Gaussian, the values $|\lambda_1 v_1^\top \theta|, |\lambda_2 v_2^\top \theta|, \ldots, |\lambda_k v_k^\top \theta|$ will be distinct with high probability.

Lemma 7.1 ([AGH+15])**.** *Let $T \in \otimes^3 \boldsymbol{R}^d$ have an orthogonal decomposition $T = \sum_{i=1}^k \lambda_i v_i \otimes v_i \otimes v_i$. For a vector $\theta_0 \in \boldsymbol{R}^d$, suppose that the set of numbers $|\lambda_1 v_1^\top \theta_0|, |\lambda_2 v_2^\top \theta_0|, \ldots, |\lambda_k v_k^\top \theta_0|$ has a unique largest element. Without loss of generality, say $|\lambda_1 v_1^\top \theta_0|$ is this largest value and $|\lambda_2 v_2^\top \theta_0|$ is the second largest value. For $t = 1, 2, \ldots$ let*

$$\theta_t := \frac{T(\cdot, \theta_{t-1}, \theta_{t-1})}{\|T(\cdot, \theta_{t-1}, \theta_{t-1})\|}. \tag{7.1}$$

Then

$$\|v_1 - \theta_t\|^2 \leq \left(2\lambda_1^2 \sum_{i=2}^k \lambda_i^{-2}\right) \cdot \left|\frac{\lambda_2 v_2^\top \theta_0}{\lambda_1 v_1^\top \theta_0}\right|^{2^{t+1}}.$$

That is, repeated iteration of 7.1 starting from θ_0 converges to v_1 at a quadratic rate.

Proof. Let $v_1, v_2, \ldots, v_d$ be an orthonormal basis and $c_i = v_i^\top \theta_0$ for $i \in [d]$, then $\theta_0 = \sum_{i=1}^d c_i v_i$. Then

$$\theta_t = \frac{\sum_{i=1}^k \lambda_i^{2^t - 1} c_i^{2^t} v_i}{\|\sum_{i=1}^k \lambda_i^{2^t - 1} c_i^{2^t} v_i\|}$$

and

$$v_1^\top \theta_t = \frac{\lambda_1^{2^t-1} c_1^{2^t}}{\|\sum_{i=1}^k \lambda_i^{2^t-1} c_i^{2^t} v_i\|} = \frac{1}{\sqrt{1 + \sum_{i=2}^k \left(\frac{\lambda_i c_i}{\lambda_1 c_1}\right)^{2^{t+1}} \left(\frac{\lambda_1}{\lambda_i}\right)^2}}. \tag{7.2}$$

At step t,

$$\begin{aligned}\|v_1 - \theta_t\|^2 = \|v_1\|^2 + \|\theta_t\|^2 - 2v_1^\top \theta_t &= 2 - \frac{2}{\sqrt{1 + \sum_{i=2}^k \left(\frac{\lambda_i c_i}{\lambda_1 c_1}\right)^{2^{t+1}} \left(\frac{\lambda_1}{\lambda_i}\right)^2}} \\ &\leq 2\sum_{i=2}^k \left(\frac{\lambda_i c_i}{\lambda_1 c_1}\right)^{2^{t+1}} \left(\frac{\lambda_1}{\lambda_i}\right)^2 \\ &\leq \left(2\lambda_1^2 \sum_{i=2}^k \lambda_i^{-2}\right) \cdot \left|\frac{\lambda_2 c_2}{\lambda_1 c_1}\right|^{2^{t+1}}.\end{aligned} \tag{7.3}$$

□

For a rank ℓ tensor with $\ell > 3$,

$$\theta_t = \frac{\sum_{i=1}^k \lambda_i^{(\ell-1)^t - 1} c_i^{(\ell-1)^t} v_i}{\|\sum_{i=1}^k \lambda_i^{(\ell-1)^t - 1} c_i^{(\ell-1)^t} v_i\|}.$$

So, the convergence rate is even faster.

To apply Tensor Power Iteration to tensors obtained from moments of samples, e.g., $T = \mathsf{E}_S(x \otimes x \otimes x)$, we don't need to compute T. Rather each step can be implemented as

$$T[\cdot, v, v] = \mathsf{E}_S(x \otimes x \otimes x)[\cdot, v, v] = \frac{1}{|S|} \sum_{x_i \in S} (x_i^\top v)^2 x_i. \tag{7.4}$$

Chapter 8

Cut decompositions

In this chapter, we study the existence and algorithms for decomposing matrices (and higher-dimensional arrays or tensors) into sums of a smaller number of particularly simple matrices called *cut* matrices. A matrix B is called a cut matrix if there exist subsets R, C of rows and columns respectively and a real number b s.t.

$$B_{ij} = \begin{cases} b & \text{if } i \in R, j \in C \\ 0 & otherwise. \end{cases}$$

Alternatively,

$$B = b(\mathbf{1}^R \otimes \mathbf{1}^C)$$

where $\mathbf{1}^R$ is the indicator vector for the set R, defined as $\mathbf{1}^R(i) = 1$ iff $i \in R$. Thus, cut matrices are rank-1 matrices with all nonzero entries being equal . A decomposition into a sum of cut matrices is called a *cut decomposition*. Cut decompositions have the following properties:

- Any matrix has an "approximate" cut decomposition with a "small" number of cut matrices.
- The decomposition can be found efficiently, in fact in constant time (implicitly) with a uniform random sample of $O(1)$ entries from the matrix.
- The decomposition allows one to solve constraint satisfaction problems up to additive error. This application, and classes of instances for which this approximation also implies a multiplicative $(1 + \epsilon)$ approximation, is discussed in the chapter on optimization using low-rank approximation.
- While the classical Singular Value Decomposition has no analog for tensors, cut decompositions and their approximation properties extend naturally to r-dimensional arrays.

In the rest of this chapter, A, B will denote $m \times n$ real matrices; S and T will be subsets of rows and columns respectively. We let

$$A(S, T) = \sum_{i \in S, j \in T} A_{ij}.$$

The *cut norm* of a matrix A is defined as

$$\|A\|_\square = \max_{S,T} |A(S,T)|.$$

Intuitively, if two matrices are close w.r.t. the cut norm, i.e., $\|A-B\|_\square$ is small, then on every cut the sum of their entries is approximately equal. In Chapter **??**, we study applications of cut decompositions to combinatorial optimization.

8.1 Existence of small cut decompositions

We begin with a proof of existence of a good cut decomposition for any matrix.

Lemma 8.1. *Let $A \in \boldsymbol{R}^{m\times n}$. For any $\epsilon > 0$, there exist $t \le 1/\epsilon^2$ cut matrices $B_1, \ldots, B_t$ whose sum $B = \sum_{i=1}^t B_t$ satisfies*

$$\|A - B\|_\square \le \epsilon\sqrt{mn}\|A\|_F.$$

We note that if $|A_{ij}| \le 1$ for all i, j, then the upper bound is ϵmn. Moreover, the inequality above is tight with the cut norm on the LHS and the Frobenius norm on the RHS (i.e., we cannot get the same norm on both sides).

Exercise 8.1. *Let $a_1, a_2, \ldots, a_n$ be real numbers and $\bar{a}$ be their average. Define $b_i = a_i - \bar{a}$. Show that*

$$\sum_{i=1}^n b_i^2 \le \sum_{i=1}^n a_i^2.$$

Proof. (of Lemma 8.1). If $\|A\|_\square \le \epsilon\sqrt{mn}\|A\|_F$, we take $B = 0$. Otherwise, for some S, T such that

$$|A(S,T)| > \epsilon\sqrt{mn}\|A\|_F.$$

Define B_1 to be the cut matrix defined by these subsets S, T with

$$(B_1)_{ij} = \frac{A(S,T)}{|S||T|}$$

for all $(i,j) \in S \times T$ and zero elsewhere. Now consider the Frobenius norm of $A - B_1$

$$\begin{aligned} \|A - B_1\|_F^2 &= \|A\|_F^2 - \|B_1\|_F^2 \\ &\le \|A\|_F^2 - \epsilon^2 \frac{mn}{|S||T|}\|A\|_F^2 \\ &\le \|A\|_F^2 - \epsilon^2\|A\|_F^2. \end{aligned}$$

We recurse on the matrix $A - B_1$ using the same condition on the cut norm. Since each step reduces the squared Frobenius norm by ϵ^2 times the initial value, it must terminate in at most $1/\epsilon^2$ steps. □

8.2 Cut decomposition algorithm

How quickly can one find a cut decomposition of a given matrix with the approximation property asserted in Lemma 8.1? In principle, this is a combinatorial problem with exponentially many candidates. From the proof of the Lemma (8.1), we see that it suffices to determine if the cut norm of A is at most ϵmn and if not, to find an S, T with $|A(S,T)| \geq \epsilon mn$. Computing the cut norm is NP-hard. However, to recover the cut decomposition guarantee, an approximate version suffices: find the maximum value of $|A(S,T)|$ to within additive error $(\epsilon/2)mn$. This reduces to two problems: $\max A(S,T)$ and $\max -A(S,T)$. We describe now an efficient algorithm for computing $\max A(S,T)$; we call this the *Maximum submatrix* problem.

For a subset of rows S, we observe that a subset of columns T that maximizes $A(S,T)$ is easy to compute, namely,

$$T = \{j : A(S,j) > 0\}.$$

We will make use of this simple observation in the algorithm.

Algorithm: Max-Submatrix

1. Pick a subset W of s rows uniformly at random.
2. For every subset $\tilde{W}$ of W, find the set of columns $\tilde{T}$ whose sum in the $\tilde{W}$ rows is positive.
3. For each candidate $\tilde{T}$, let $\tilde{S}$ be the rows with positive sum in the $\tilde{T}$ columns.
4. Among all candidate $\tilde{S}, \tilde{T}$ enumerated above, output the subsets that achieve $\max A(\tilde{S}, \tilde{T})$.

The idea behind the algorithm is the following: we guess a small random subset of the optimal subset of rows S, then use this small subset to approximately identify the columns with positive sums. A small random subset should suffice since this will suffice to identify columns with significant sums, and the ones with sums close to zero are relatively unimportant. We guess such a subset by first picking a random subset of rows W of the full matrix and enumerating all possibilities for $S \cap W$, i.e., all subsets of W. In the estimating column sums step, if we had the correct $S \cap W$, (which will happen at some point in the enumeration) the only columns on which we could be wrong about the sign of the column sum in the S rows are ones where the column sum is close to zero, and these do not contribute significantly to $A(S,T)$. So, one of our candidates $\tilde{T}$ is approximately the best one in the sense that $A(S,\tilde{T})$ is high for the optimal row subset S.

Now we turn to a rigorous analysis of the algorithm. It is easy to see that the running time of the algorithm is $O(mn2^s)$. Setting $s = 16/\epsilon^2$, the next theorem

recovers the additive guarantee of the existence lemma in time $O(mn2^s) = O(mn2^{16/\epsilon^2})$. In the next section, we will see how the running time can be further improved.

Theorem 8.2. *With probability at least $1-\delta$, the subsets S, T found by Algorithm Max-Submatrix we satisfy*

$$A(S', T') \geq A(S^*, T^*) - \frac{1}{\delta}\sqrt{\frac{mn}{s}}\|A\|_F$$

where S^*, T^* is an optimal solution.

The proof of this theorem relies on the following lemma.

Lemma 8.3. *Suppose A is an $m \times n$ matrix. Fix a subset S of rows of A. Let T be the set of columns with positive sum in the S rows. Let W be a uniform random subset of rows of cardinality s and $\tilde{T}$ be the set of columns with positive sum in the $W \cap S$ rows. Then,*

$$\mathsf{E}_W(A(S, \tilde{T})) \geq A(S, T) - \sqrt{\frac{mn}{s}}\|A\|_F. \tag{8.1}$$

The following lemma will be useful in the proof.

Lemma 8.4. *Let $X_1, \ldots, X_n$ be i.i.d. random variables from some distribution and $Y_1, \ldots, Y_n$ be random variables from the same distribution but without replacement, i.e., Y_j is drawn from the original distribution restricted to the complement of $Y_1, \ldots, Y_{j-1}$. Then for any convex function $\phi : \boldsymbol{R} \to \boldsymbol{R}$,*

$$\mathsf{E}(\phi(\sum_{i=1}^{n} Y_i)) \leq \mathsf{E}(\phi(\sum_{i=1}^{n} X_i)).$$

Proof. (of Lemma 8.3). We write

$$A(S, \tilde{T}) = A(S, T) - A(S, B_1) + A(S, B_2), \tag{8.2}$$

where

$$\begin{aligned} B_1 &= \{j : A(S, j) > 0 \text{ and } A(S \cap W, j) \leq 0\}, \\ B_2 &= \{j : A(S, j) \leq 0 \text{ and } A(S \cap W, j) > 0\}. \end{aligned}$$

Let $a(j) = \sum_{i \in S} A_{ij}^2$,

$$X_j = A(S \cap W, j) = \sum_{i \in S} A_{ij} \mathbf{1}^W(i)$$

where $\mathbf{1}^W$ is the indicator vector of the subset W. Therefore,

$$\mathsf{E}(X_j) = \frac{s}{m} A(S, j).$$

To bound the variance we will use Lemma 8.4 with the convex function $\phi(x) = x^2$.

$$\mathsf{Var}\,(X_j) \leq \mathsf{E}\,((\sum_{i\in S} A_{ij}\mathbf{1}^W(i))^2) \leq \frac{s}{m}a(j).$$

Hence, for any $t \geq 0$, using Chebychev's inequality,

$$\mathbf{Pr}\left(\left|X_j - \frac{s}{m}A(S,j)\right| \geq t\right) \leq \frac{sa(j)}{mt^2} \tag{8.3}$$

If $j \in B_1$ then

$$X_j - \frac{s}{m}A(S,j) \leq -\frac{s}{m}A(S,j)$$

and so applying (8.3) with

$$t = \frac{s}{m}A(S,j),$$

we get that for each fixed j,

$$\mathbf{Pr}(j \in B_1) \leq \frac{ma(j)}{sA(S,j)^2}.$$

Thus,

$$\begin{aligned}\mathsf{E}\left(\sum_{j\in B_1} A(S,j)\right) &\leq \sum_{\{j:\ A(S,j)>0\}} \min\left\{A(S,j), \frac{ma(j)}{sA(S,j)}\right\} \\ &\leq \sum_{\{j:\ A(S,j)>0\}} \sqrt{\frac{ma(j)}{s}} \end{aligned} \tag{8.4}$$

By an identical argument we obtain

$$\mathsf{E}\left(\sum_{j\in B_2} A(S,j)\right) \geq -\sum_{\{j:\ A(S,j)<0\}} \sqrt{\frac{ma(j)}{s}}.$$

Hence, using Cauchy-Schwartz,

$$\mathsf{E}\,(A(S,\tilde{T})) \geq A(S,T) - \sum_j \sqrt{\frac{ma(j)}{s}} \geq A(S,T) - \sqrt{\frac{mn}{s}}\|A\|_F.$$

□

We conclude this section with a proof of the main guarantee for the algorithm.

Proof. (of Theorem 8.2.) Let $S^*, T^* = \arg\max A(S,T)$ where the maximum is over all choices of subsets S, T. Let $\tilde{S}, \tilde{T}$ be the subsets found by the algorithm. Applying Lemma 8.3 to S^*, T^*, we have

$$\mathsf{E}\,(A(\tilde{S},\tilde{T})) \geq A(S^*,T^*) - \sqrt{\frac{mn}{s}}\|A\|_F.$$

Alternatively,

$$\mathsf{E}\left(A(S^*,T^*) - A(\tilde{S},\tilde{T})\right) \leq \sqrt{\frac{mn}{s}}\|A\|_F.$$

Applying Markov's inequality, for any $0 \leq \delta \leq 1$,

$$\Pr(A(S^*,T^*) - A(\tilde{S},\tilde{T}) \geq \frac{1}{\delta}\sqrt{\frac{mn}{s}}\|A\|_F) \leq \delta.$$

In other words, with probability at least $1-\delta$,

$$A(\tilde{S},\tilde{T}) \geq A(S^*,T^*) - \frac{1}{\delta}\sqrt{\frac{mn}{s}}\|A\|_F.$$

□

8.3 A constant-time algorithm

In this section, we extend Algorithm Max-Submatrix so that in *constant time*, it finds an implicit representation of S, T that approximately maximize $A(S,T)$. By constant time we mean a time bound that depends only on ϵ and not on m, n.

The idea is simple: Pick uniform random subsets $\hat{S}$ of $\hat{s} = O(1/\epsilon^4)$ rows and $\hat{T}$ of $\hat{s}$ columns at the outset. Instead of A, we use this sampled submatrix for all row and column sum estimates, i.e., we estimate $A(\tilde{S},\tilde{T})$ by

$$\frac{mn}{\hat{s}^2}A(\tilde{S}\cap\hat{S},\tilde{T}\cap\hat{T}),$$

for which we only need to know the entries of A in $\hat{S}\times\hat{T}$. One can show using the Höffding-Azuma inequality that the estimate is within additive error at most $O(\epsilon mn)$ with high probability.

Theorem 8.5. *For A with $|A_{ij}| \leq 1$, for any fixed $\epsilon > 0$, Algorithm Max-Submatrix executed with $s = O(1/\epsilon^2)$ on a random submatrix of A induced by $\hat{s}$ random rows and $\hat{s}$ random columns of A finds $\tilde{S},\tilde{T}$ such that with probability at least $3/4$,*

$$|\max A(S,T) - \frac{mn}{\hat{s}^2}A(\tilde{S},\tilde{T})| \leq \frac{16mn}{\sqrt{s}}.$$

Moreover, the running time to find $\tilde{S},\tilde{T}$, from which the approximately optimal S,T can be found is 2^s. A full cut-decomposition B with the guarantees of Lemma 8.1 can be implicitly computed in time $O(s^3 2^s)$.

Thus, the *value* of $\max A(S,T)$ can be estimated in *constant* time independent of m, n.

Exercise 8.2. *Prove Theorem 8.5 for $\hat{s} = O(1/\epsilon^4)$, by showing that with probability at least 7/8, for every $\tilde{T} \subset T$,*

$$|A(\tilde{S}, \tilde{T}) - \frac{mn}{\hat{s}^2} A(\tilde{S} \cap \hat{S}, \tilde{T} \cap \hat{T})| \leq \epsilon mn.$$

[Hint: use the Höeffding-Azuma inequality for any candidate $\tilde{T}$ and apply a union bound. The inequality says that given random variables $X_1, \ldots, X_n$ with $|X_i| \leq 1$ and $\mathsf{E}(X_i | X_1, \ldots, X_{i-1}) = 0$ for all $i \in [n]$, their sum X satisfies

$$\mathbf{Pr}(|X - \mathsf{E}(X)| \geq t) \leq 2e^{-\frac{t^2}{2n}}.]$$

8.4 Cut decompositions for tensors

We next consider r-dimensional tensors, i.e., $A \in \boldsymbol{R}^{n_1 \times n_2 \times \ldots \times n_r}$. For subsets $S_1 \subseteq [n_1], \ldots, S_r \subseteq [n_r]$ of the indices, we can define $A(S_1, \ldots, S_r)$ as the sum of all entries in the induced subtensor. The cut norm generalizes as follows:

$$\|A\|_\square = \max |A(S_1, \ldots, S_r)|.$$

A cut tensor is tensor defined by a subsets of indices $S_1, \ldots, S_r$, with $A_{i_1,\ldots,i_r}$ being a constant if $(i_1, \ldots, i_r) \in S_1 \times \ldots \times S_r$ and zero otherwise. A cut tensor is thus a rank-one tensor obtained as the outer product of r vectors with entries from $\{0, 1\}$. The existence of cut decompositions is captured in the next exercise.

Exercise 8.3. *Show that Lemma 8.1 extends to r-dimensional tensors for any r, i.e., if $A \in \boldsymbol{R}^{n_1 \times n_2 \times \ldots \times n_r}$, for any $\epsilon > 0$, there exist $t \leq 1/\epsilon^2$ cut tensors $B_1, \ldots, B_t$ s.t. their sum $B = \sum_{i=1}^t B_i$ satisfies*

$$\|A - B\|_\square \leq \epsilon \sqrt{n_1 n_2 \ldots n_r} \|A\|_F.$$

The idea for solving the maximum subtensor problem to within additive error is the following. First we observe that given subsets $S_1, \ldots, S_{r-1}$, we can find the optimal subset $S_r \subseteq [n_r]$ as the subset of indices with positive sums, i.e., i s.t. $A(S_1, \ldots, S_{r-1}, i)$ is positive. Using this,

1. We fix random subsets $W_1, W_2, \ldots, W_{r-1}$ each of size s.
2. We enumerate all possible subsets $\tilde{W}_t \subseteq W_t$ as candidates for $S_t \cap W_t$. For each such candidate $\tilde{W}_1, \ldots, \tilde{W}_{r-1}$, we find the best subset $\tilde{S}_r$.
3. For each $\tilde{S}_r$, we form the $(r-1)$-tensor $\tilde{A}$ as

$$\tilde{A}_{i_1 i_2 \ldots i_{r-1}} = \sum_{i \in S_r} A_{i_1 i_2 \ldots i_{r-1} i}$$

   and recursively solve the maximum subtensor problem for this $r-1$ tensor.
4. Among all the candidates enumerated, choose the best $\tilde{S}_r$.

Exercise 8.4. *Show that in order to achieve the guarantee of Exercise 8.3 with high probability, it suffices to set $s = poly(r, 1/\epsilon)$.*

8.5 A weak regularity lemma

In this section, we study a variant of the regularity lemma from graph theory. We first briefly recall Szemerédi's original lemma.

Let G be an undirected graph on n vertices. For a pair of subset of vertices A, B, let $e(A, B)$ be the number of edges between A and B (A and B could intersect). The density of a pair A, B is

$$d(A, B) = \frac{e(A, B)}{|A||B|}.$$

A pair of subsets $V_1, V_2 \subseteq V$ is said to be ϵ-regular if for any two subsets $A \subseteq V_1, B \subseteq V_2$,

$$|d(A, B) - d(V_1, V_2)| \leq \epsilon.$$

A partition of V into k subsets is called an ϵ-regular partition if all but ϵk^2 pairs of subsets are ϵ-regular.

Theorem 8.6 (Regularity)**.** *For any $\epsilon > 0$, there exists $k = k(\epsilon)$, such that for any graph G there is an ϵ-regular partition with at most k parts.*

The theorem can be further strengthened by noting that no part is too large and the part sizes are comparable. This powerful theorem has many applications in combinatorics, analysis and computer science. Unfortunately, the function $k(\epsilon)$ is a tower function of height $1/\epsilon$ and such a dependence is unavoidable in general.

The weak regularity lemma, which we state next, follows from cut decompositions and has a much smaller bound on the size of a partition, albeit with a weaker guarantee.

Theorem 8.7 (Weak Regularity)**.** *For any $\epsilon > 0$, and any graph G, there exists a partition of the vertices of G into $k \leq 2^{2/\epsilon^2}$ parts such that for any two subset $S, T \subseteq V$,*

$$\left| e(S, T) - \sum_{i,j=1}^{k} d(V_i, V_j)|S \cap V_i||T \cap V_j|) \right| \leq \epsilon n^2.$$

Such a parition is called an ϵ-pseudo-regular partition of the graph.

Proof. Let $B_1, \ldots, B_s$ be a cut decomposition of the adjacency matrix of G. Consider the partition induced by the subsets $R_1, \ldots, R_s, T_1, \ldots, T_s$ of size $k \leq 2^{2s}$. The matrix $B = B_1 + \ldots + B_s$ can be partitioned into a sum of k cut matrices with disjoint support so that B is their sum. Moreoever the value of each cut matrix is simply its density in the original graph. This completes the proof. □

8.6 Discussion

Szemerédi

Nesterov, Alon-Naor Grothendieck approximations

Cut decompositions were introduced by Frieze and Kannan [FK99].

Graph limits

Chapter 9

Matrix approximation by Random Sampling

In this chapter, we study randomized algorithms for matrix multiplication and low-rank matrix and tensor approximation. The main motivation is to obtain efficient approximations using only randomly sampled subsets of given matrices. We remind the reader that for a vector-valued random variable X, we write $\mathsf{Var}\,(X) = \mathsf{E}\,(\|X - \mathsf{E}\,(X)\|^2)$ and similarly for a matrix-valued random variable, with the norm denoting the Frobenius norm in the latter case.

9.1 Matrix-vector product

In many numerical algorithms, a basic operation is the matrix-vector product. If A is a $m \times n$ matrix and v is an n vector, we have ($A^{(j)}$ denotes the j'th column of A):

$$Av = \sum_{j=1}^{n} A^{(j)} v_j.$$

The right-hand side is the sum of n vectors and can be estimated by using a sample of the n vectors. The error is measured by the variance of the estimate. It is easy to see that a uniform random sample could have high variance — consider the example when only one column is nonzero.

This leads to the question: what distribution should the sample columns be chosen from? Let $p_1, p_2, \ldots p_n$ be nonnegative reals adding up to 1. Pick $j \in \{1, 2, \ldots n\}$ with probability p_j and consider the vector-valued random variable

$$X = \frac{A^{(j)} v_j}{p_j}.$$

Clearly $\mathsf{E}\, X = Av$, so X is an unbiased estimator of Av. We also get

$$\mathsf{Var}\,(X) = \mathsf{E}\,\|X\|^2 - \|\mathsf{E}\,X\|^2 = \sum_{j=1}^{n} \frac{\|A^{(j)}\|^2 v_j^2}{p_j} - \|Av\|^2. \tag{9.1}$$

Now we introduce an important probability distribution on the columns of a matrix A, namely the **length-squared** (LS) distribution, where a column is picked with probability proportional to its squared length. We will say

$$j \text{ is drawn from } \mathrm{LS}_{\mathrm{col}}(A) \quad \text{if} \quad p_j = \|A^{(j)}\|^2/\|A\|_F^2.$$

This distribution has useful properties. An *approximate* version of this distribution - $\mathrm{LS}_{\mathrm{col}}(A, c)$, where we only require that

$$p_j \geq c\|A^{(j)}\|^2/\|A\|_F^2$$

for some $c \in (0, 1)$ also shares interesting properties. If j is from $\mathrm{LS}_{col}(A, c)$, then note that the expression (9.1) simplifies to yield

$$\mathsf{Var}\, X \leq \frac{1}{c}\|A\|_F^2\|v\|^2.$$

Taking the average of s i.i.d. trials decreases the variance by a factor of s. So, if we take s independent samples $j_1, j_2, \ldots j_s$ (i.i.d., each picked according to $\mathrm{LS}_{\mathrm{col}}(A, c)$), then with

$$Y = \frac{1}{s}\sum_{t=1}^{s} \frac{A^{(j_t)} v_{j_t}}{p_{j_t}},$$

we have

$$\mathsf{E}\, Y = Av$$

and

$$\mathsf{Var}\, Y = \frac{1}{s}\sum_{j} \frac{\|A^{(j)}\|^2 v_j^2}{p_j} - \frac{1}{s}\|Av\|^2 \leq \frac{1}{cs}\|A\|_F^2\|v\|^2. \tag{9.2}$$

Such an approximation for matrix vector products is useful only when $\|Av\|$ is comparable to $\|A\|_F\|v\|$. It is greater value for matrix multiplication.

In certain contexts, it may be easier to sample according to $\mathrm{LS}(A, c)$ than the exact length squared distribution. We have used the subscript $_{\mathrm{col}}$ to denote that we sample columns of A; it will be sometimes useful to sample rows, again with probabilities proportional to the length squared (of the row, now). In that case, we use the subscript $_{\mathrm{row}}$.

9.2 Matrix Multiplication

The next basic problem is that of multiplying two matrices, A, B, where A is $m \times n$ and B is $n \times p$. From the definition of matrix multiplication, we have

$$AB = \left(AB^{(1)}, AB^{(2)}, \ldots AB^{(p)}\right).$$

Applying (9.2) p times and adding, we get the next theorem (recall the notation that $B_{(j)}$ denotes row j of B).

Theorem 9.1. *Let $p_1, p_2, \ldots p_n$ be non-negative reals summing to 1 and let $j_1, j_2, \ldots j_s$ be i.i.d. random variables, where j_t is picked to be one of $\{1, 2, \ldots n\}$ with probabilities $p_1, p_2, \ldots p_n$ respectively. Then with*

$$\begin{aligned} Y &= \frac{1}{s}\sum_{t=1}^{s}\frac{A^{(j_t)}B_{(j_t)}}{p_{j_t}}, \\ \mathsf{E}\, Y &= AB \qquad \text{and} \qquad \mathsf{Var}\, Y = \frac{1}{s}\sum_{j=1}^{n}\frac{\|A^{(j)}\|^2\|B_{(j)}\|^2}{p_j} - \|AB\|_F^2. \end{aligned} \tag{9.3}$$

If j_t are distributed according to $LS_{col}(A, c)$, then

$$\mathsf{Var}\, Y \leq \frac{1}{cs}\|A\|_F^2\|B\|_F^2.$$

A special case of matrix multiplication which is both theoretically and practically useful is the product AA^T.

The singular values of AA^T are just the squares of the singular values of A. So it can be shown that if $B \approx AA^T$, then the eigenvalues of B will approximate the squared singular values of A. Later, we will want to approximate A itself well. For this, we will need in a sense a good approximation to not only the singular values, but also the singular vectors of A. This is a more difficult problem. However, approximating the singular values well via AA^T will be a crucial starting point for the more difficult problem.

For the matrix product AA^T, the expression for $\mathsf{Var}\, Y$ (in (9.3)) simplifies to

$$\mathsf{Var}\, Y = \frac{1}{s}\sum_{j}\frac{\|A^{(j)}\|^4}{p_j} - \|AA^T\|_F^2.$$

The second term on the right-hand side is independent of p_j. The first term is minimized when the p_j conform to the length-squared distribution.

9.3 Low-rank approximation

When $B = A^T$, we may rewrite the expression (9.3) as

$$Y = CC^T, \quad \text{where, } C = \frac{1}{\sqrt{s}}\left(\frac{A^{(j_1)}}{\sqrt{p_{j_1}}}, \frac{A^{(j_2)}}{\sqrt{p_{j_2}}}, \ldots \frac{A^{(j_s)}}{\sqrt{p_{j_s}}}\right)$$

and the next theorem follows.

Theorem 9.2. *Let A be an $m \times n$ matrix and $j_1, j_2, \ldots j_s$ be i.i.d. samples from $\{1, 2, \ldots n\}$, each picked according to probabilities $p_1, p_2, \ldots p_n$. Define*

$$C = \frac{1}{\sqrt{s}}\left(\frac{A^{(j_1)}}{\sqrt{p_{j_1}}}, \frac{A^{(j_2)}}{\sqrt{p_{j_2}}}, \ldots \frac{A^{(j_s)}}{\sqrt{p_{j_s}}}\right).$$

Then,

$$\mathsf{E}\, CC^T = AA^T \qquad \textit{and} \qquad \mathsf{E}\, \|CC^T - AA^T\|_F^2 = \frac{1}{s}\sum_{j=1}^{n} \frac{|A^{(j)}|^4}{p_j} - \frac{1}{s}\|AA^T\|_F^2.$$

If the p_j's conform to the approximate length squared distribution $LS_{col}(A, c)$, then

$$\mathsf{E}\, \|CC^T - AA^T\|_F^2 \leq \frac{1}{cs}\|A\|_F^4.$$

The fact that $\|CC^T - AA^T\|_F$ is small implies that the singular values of A are close to the singular values of C. Indeed the Hoffman-Wielandt inequality asserts that

$$\sum_t \left(\sigma_t(CC^T) - \sigma_t(AA^T)\right)^2 \leq \|CC^T - AA^T\|_F^2. \tag{9.4}$$

(Exercise 9.3 asks for a proof of this inequality.)

To obtain a good low-rank approximation of A, we will also need a handle on the singular vectors of A. A natural question is whether the columns of C already contain a good low-rank approximation to A. To this end, first observe that if $u^{(1)}, u^{(2)}, \ldots u^{(k)}$ are orthonormal vectors in $\mathbf{R}^m$, then

$$\sum_{t=1}^{k} u^{(t)} u^{(t)^T} A$$

is the projection of A into the space H spanned by $u^{(1)}, u^{(2)}, \ldots u^{(k)}$, namely

(i) For any $u \in H$, $u^T A = u^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A$ and

(ii) For any $u \in H^{\perp}$, $u^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A = 0$.

This motivates the following algorithm for low-rank approximation.

Algorithm: Fast-SVD

1. Sample s columns of A from the squared length distribution to form a matrix C.
2. Find $u^{(1)}, \ldots, u^{(k)}$, the top k left singular vectors of C.
3. Output $\sum_{t=1}^{k} u^{(t)} u^{(t)^T} A$ as a rank-k approximation to A.

The running time of the algorithm (if it uses s samples) is $O(ms^2)$.

We now state and prove the main lemma of this section. Recall that A_k stands for the best rank-k approximation to A (in Frobenius norm and 2-norm) and is given by the first k terms of the SVD.

Lemma 9.3. *Suppose A, C are $m \times n$ and $m \times s$ matrices respectively with $s \leq n$ and U is the $m \times k$ matrix consisting of the top k singular vectors of C. Then,*

$$\begin{aligned} \|A - UU^T A\|_F^2 &\leq \|A - A_k\|_F^2 + 2\sqrt{k}\|AA^T - CC^T\|_F \\ \|A - UU^T A\|_2^2 &\leq \|A - A_k\|_2^2 + \|CC^T - AA^T\|_2 + \|CC^T - AA^T\|_F. \end{aligned}$$

Proof. We have

$$\|A - \sum_{t=1}^{k} u^{(t)} {u^{(t)}}^T A\|_F^2 = \|A\|_F^2 - \|U^T A\|_F^2$$

and

$$\|C_k\|_F^2 = \|U^T C\|_F^2.$$

Using these equations,

$$\begin{aligned} & \|A - \sum_{t=1}^{k} u^{(t)} {u^{(t)}}^T A\|_F^2 - \|A - A_k\|_F^2 \\ = \; & \|A\|_F^2 - \|U^T A\|_F^2 - (\|A\|_F^2 - \|A_k\|_F^2) \\ = \; & \left(\|A_k\|_F^2 - \|C_k\|_F^2\right) + \|U^T C\|_F^2 - \|U^T A\|_F^2 \\ = \; & \sum_{t=1}^{k} \left(\sigma_t(A)^2 - \sigma_t(C)^2\right) + \sum_{t=1}^{k} \left(\sigma_t(C)^2 - \|{u^{(t)}}^T A\|^2\right) \\ \leq \; & \sqrt{k \sum_{t=1}^{k} \left(\sigma_t(A)^2 - \sigma_t(C)^2\right)^2} + \sqrt{k \sum_{t=1}^{k} \left(\sigma_t(C)^2 - \|{u^{(t)}}^T A\|^2\right)^2} \\ = \; & \sqrt{k \sum_{t=1}^{k} \left(\sigma_t(AA^T) - \sigma_t(CC^T)\right)^2} + \sqrt{k \sum_{t=1}^{k} \left({u^{(t)}}^T (CC^T - AA^T) u^{(t)}\right)^2} \\ \leq \; & 2\sqrt{k}\|AA^T - CC^T\|_F. \end{aligned}$$

Here we first used the Cauchy-Schwarz inequality on both summations and then the Hoffman-Wielandt inequality 9.4.

The proof of the second statement also uses the Hoffman-Wielandt inequality. □

Exercise 9.1. *Prove the 2-norm bound in the statement of Lemma 9.3.*

We can now combine Theorem 9.2 and Lemma 9.3 to obtain the main theorem of this section.

Theorem 9.4. *Algorithm Fast-SVD finds a rank-k matrix $\tilde{A}$ such that*

$$\mathsf{E}\left(\|A - \tilde{A}\|_F^2\right) \leq \|A - A_k\|_F^2 + 2\sqrt{\frac{k}{s}}\|A\|_F^2$$

$$\mathsf{E}\left(\|A - \tilde{A}\|_2^2\right) \leq \|A - A_k\|_2 + \frac{2}{\sqrt{s}}\|A\|_F^2.$$

Exercise 9.2. *Using the fact that* $\|A\|_F^2 = Tr(AA^T)$ *show that:*

1. *For any two matrices* P, Q*, we have* $|TrPQ| \leq \|P\|_F \|Q\|_F$*.*

2. *For any matrix* Y *and any symmetric matrix* X*,* $|TrXYX| \leq \|X\|_F^2 \|Y\|_F$*.*

Exercise 9.3. *Prove the Hoffman-Wielandt inequality for symmetric matrices: for any two* $n \times n$ *symmetric matrices* A *and* B*,*

$$\sum_{t=1}^{n} (\sigma_t(A) - \sigma_t(B))^2 \leq \|A - B\|_F^2.$$

(Hint: consider the SVD of both matrices and note that any doubly stochastic matrix is a convex combination of permutation matrices).

Exercise 9.4. *(Sampling on the fly) Suppose you are reading a list of real numbers* $a_1, a_2, \ldots a_n$ *in a streaming fashion, i.e., you only have* $O(1)$ *memory and the input data comes in arbitrary order in a stream. Your goal is to output a number* X *between* 1 *and* n *such that:*

$$Pr(X = i) = \frac{a_i^2}{\sum_{j=1}^{n} a_j^2}.$$

How would you do this? How would you pick values for $X_1, X_2, \ldots X_s$ *(*$s \in O(1)$*) where the* X_i *are i.i.d.?*

In this section, we considered projection to the span of a set of orthogonal vectors (when the $u^{(t)}$ form the top k left singular vectors of C). In the next section, we will need to deal also with the case when the $u^{(t)}$ are not orthonormal. A prime example we will deal with is the following scenario: suppose C is an $m \times s$ matrix, for example obtained by sampling s columns of A as above. Now suppose $v^{(1)}, v^{(2)}, \ldots v^{(k)}$ are indeed an orthonormal set of vectors for which $C \approx C \sum_{t=1}^{k} v^{(t)} v^{(t)^T}$; i.e., $\sum_{t=1}^{k} v^{(t)} v^{(t)^T}$ is a "good right projection" space for C. Then suppose the $u^{(t)}$ are defined by $u^{(t)} = Cv^{(t)}/|Cv^{(t)}|$. We will see later that $C \approx \sum_{t=1}^{k} u^{(t)} u^{(t)^T} C$; i.e., that $\sum_{t=1}^{k} u^{(t)} u^{(t)^T}$ is a good left projection space for C. The following lemma which generalizes some of the arguments we have used here will be useful in this regard.

Lemma 9.5. *Suppose* $u^{(1)}, u^{(2)}, \ldots u^{(k)}$ *are any* k *vectors in* $\mathbf{R}^m$*. Suppose* A, C *are any two matrices, each with* m *rows (and possibly different numbers of*

columns.) Then, we have

$$\begin{aligned} & \|A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\|_F^2 - \|C - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} C\|_F^2 \\ \leq \quad & \|A\|_F^2 - \|C\|_F^2 \\ + \quad & \|AA^T - CC^T\|_F \|\sum_{t=1}^{k} u^{(t)} u^{(t)^T}\|_F \left(2 + \|\sum_{t=1}^{k} u^{(t)} u^{(t)^T}\|_F\right) \qquad (9.5) \\ & \|A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\|_2^2 - \|C - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} C\|_2^2 \\ \leq \quad & \|AA^T - CC^T\|_2 \left(\|\sum_{t=1}^{k} u^{(t)} u^{(t)^T}\|_2 + 1\right)^2. \qquad (9.6) \end{aligned}$$

Proof.

$$\begin{aligned} & \|A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\|_F^2 \\ = \quad & \mathrm{Tr}\left((A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A)(A^T - A^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T})\right) \\ = \quad & \mathrm{Tr} AA^T + \mathrm{Tr} \sum_{t=1}^{k} u^{(t)} u^{(t)^T} AA^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} - 2\mathrm{Tr} \sum_{t=1}^{k} u^{(t)} u^{(t)^T} AA^T, \end{aligned}$$

where we have used the fact that square matrices commute under trace. We do the same expansion for C to get

$$\begin{aligned} & \|A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\|_F^2 - \|C - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} C\|_F^2 - \left(\|A\|_F^2 - \|C\|_F^2\right) \\ = \quad & \mathrm{Tr} \sum_{t=1}^{k} u^{(t)} u^{(t)^T} (AA^T - CC^T) \sum_{t=1}^{k} u^{(t)} u^{(t)^T} - 2\mathrm{Tr} \sum_{t=1}^{k} u^{(t)} u^{(t)^T} (AA^T - CC^T) \\ \leq \quad & \|\sum_{t=1}^{k} u^{(t)} u^{(t)^T}\|_F^2 \|AA^T - CC^T\|_F + 2\|\sum_{t=1}^{k} u^{(t)} u^{(t)^T}\|_F \|AA^T - CC^T\|_F, \end{aligned}$$

where we have used two standard inequalities: $|\mathrm{Tr}PQ| \leq \|P\|_F \|Q\|_F$ for any matrices P, Q and $|\mathrm{Tr}XYX| \leq \|X\|_F^2 \|Y\|_F$ for any Y and a symmetric matrix X (see Exercise 9.2). This gives us (9.5).

For (9.6), suppose v is the unit length vector achieving

$$\|v^T (A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A)\| = \|A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\|_2.$$

Then we expand

$$\begin{aligned} & \|v^T(A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A)\|^2 \\ = \quad & v^T(A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A)(A^T - A^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T})v \\ = \quad & v^T A A^T v - 2v^T A A^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} v + v^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A A^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} v, \end{aligned}$$

and the corresponding terms for C. Now, (9.6) follows by a somewhat tedious but routine calculation. □

9.3.1 A sharper existence theorem

In this section, we establish the existence of a rank k approximation of any matrix A in the span of a small sample of columns from $LS_{col}(A)$. The bound will be better than the one achieved in the algorithm above, namely for additive error $\epsilon\|A\|_F^2$, one needs only $O(k/\epsilon)$ columns rather than $O(k/\epsilon^2)$ columns. We will see later that this bound is asymptotically optimal.

Theorem 9.6. *Let S be an i.i.d. sample of s columns from $LS_{col}(A)$. There exists a matrix $\tilde{A}$ of rank at most k, with columns in the span of S s.t.,*

$$\begin{aligned} \mathsf{E}\left(\|A - \tilde{A}\|_F^2\right) &\leq \|A - A_k\|_F^2 + \frac{k}{s}\|A\|_F^2 \\ \mathsf{E}\left(\|A - \tilde{A}\|_2^2\right) &\leq \|A - A_k\|_2^2 + \frac{1}{s}\|A\|_F^2 \end{aligned}$$

where A_k is the best rank-k approximation of A as given by its SVD.

9.4 Invariant subspaces

The classical SVD has associated with it the decomposition of space into the sum of **invariant subspaces**.

Theorem 9.7. *Let A be a $m \times n$ matrix and $v^{(1)}, v^{(2)}, \ldots v^{(n)}$ an orthonormal basis for $\mathbf{R}^n$. Suppose for $k, 1 \leq k \leq$ rank(A) we have*

$$|Av^{(t)}|^2 = \sigma_t^2(A), \quad \text{for } t = 1, 2, , \ldots k.$$

Then

$$u^{(t)} = \frac{Av^{(t)}}{|Av^{(t)}|}, \quad \text{for } t = 1, 2, \ldots k$$

form an orthonormal family of vectors. The following hold:

$$\begin{aligned}
\sum_{t=1}^{k} |u^{(t)^T} A|^2 &= \sum_{t=1}^{k} \sigma_t^2 \\
\|A - A\sum_{t=1}^{k} v^{(t)} v^{(t)^T}\|_F^2 &= \|A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\|_F^2 \\
&= \sum_{t=k+1}^{n} \sigma_t^2(A) \\
\|A - A\sum_{t=1}^{k} v^{(t)} v^{(t)^T}\|_2 &= \|A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\|_2 = \sigma_{k+1}(A).
\end{aligned}$$

Given the right singular vectors $v^{(t)}$, a family of left singular vectors $u^{(t)}$ may be found by just applying A to them and scaling to length 1. The orthogonality of the $u^{(t)}$ is automatically ensured. So we get that given the optimal k dimensional "right projection" $A\sum_{t=1}^{k} v^{(t)} v^{(t)^T}$, we also can get the optimal "left projection"

$$\sum_{t=1}^{k} u^{(t)} u^{(t)^T} A.$$

Counting dimensions, it also follows that for any vector w orthogonal to such a set of $v^{(1)}, v^{(2)}, \ldots v^{(k)}$, we have that Aw is orthogonal to $u^{(1)}, u^{(2)}, \ldots u^{(k)}$. This yields the standard decomposition into the direct sum of subspaces.

Exercise 9.5. *Prove Theorem 9.7.*

We now extend the previous theorem to *approximate* invariance, i.e., even if the hypothesis of the previous theorem $|Av^{(t)}|^2 = \sigma_t^2(A)$ is only approximately satisfied, an approximate conclusion follows. We give below a fairly clean statement and proof formalizing this intuition. It will be useful to define the error measure

$$\Delta(A, v^{(1)}, v^{(2)}, \ldots v^{(k)}) = \mathrm{Max}_{1 \le t \le k} \sum_{i=1}^{t} (\sigma_i^2(A) - |Av^{(i)}|^2). \tag{9.7}$$

Theorem 9.8. *Let A be a matrix of rank r and $v^{(1)}, v^{(2)}, \ldots v^{(r)}$ be an orthonormal set of vectors spanning the row space of A (so that $\{Av^{(t)}\}$ span the column space of A). Then, for $t, 1 \le t \le r$, we have*

$$\begin{aligned}
&\sum_{s=t+1}^{r} \left(v^{(t)^T} A^T A v^{(s)}\right)^2 \\
\le\ & |Av^{(t)}|^2 \left(\sigma_1^2(A) + \sigma_2^2(A) + \ldots \sigma_t^2(A) - |Av^{(1)}|^2 - |Av^{(2)}|^2 - \ldots |Av^{(t)}|^2\right).
\end{aligned}$$

Note that ${v^{(t)}}^T A^T A v^{(s)}$ is the (t,s) th entry of the matrix $A^T A$ when written with respect to the basis $\{v^{(t)}\}$. So, the quantity $\sum_{s=t+1}^{r} \left({v^{(t)}}^T A^T A v^{(s)}\right)^2$ is the sum of squares of the above diagonal entries of the t th row of this matrix. Theorem (9.8) implies the classical Theorem (9.7) : $\sigma_t(A) = |Av^{(t)}|$ implies that the right hand side of the inequality above is zero. Thus, ${v^{(t)}}^T A^T A$ is colinear with ${v^{(t)}}^T$ and so $|{v^{(t)}}^T A^T A| = |Av^{(t)}|^2$ and so on.

Proof. First consider the case when $t = 1$. We have

$$\begin{aligned}\sum_{s=2}^{r} ({v^{(1)}}^T A^T A v^{(s)})^2 &= |{v^{(1)}}^T A^T A|^2 - ({v^{(1)}}^T A^T A v^{(1)})^2 \\ &\leq |Av^{(1)}|^2 \sigma_1(A)^2 - |Av^{(1)}|^4 \\ &\leq |Av^{(1)}|^2 (\sigma_1(A)^2 - |Av^{(1)}|^2). \qquad (9.8)\end{aligned}$$

The proof of the theorem will be by induction on the rank of A. If $r = 1$, there is nothing to prove. Assume $r \geq 2$. Now, Let

$$A' = A - Av^{(1)}{v^{(1)}}^T.$$

A' is of rank $r-1$. If $w^{(1)}, w^{(2)}, \ldots$ are the right singular vectors of A', they are clearly orthogonal to $v^{(1)}$. So we have for any s, $1 \leq s \leq r-1$,

$$\begin{aligned}&\sigma_1^2(A') + \sigma_2^2(A') + \ldots \sigma_s^2(A') = \sum_{t=1}^{s} |A'w^{(t)}|^2 = \sum_{t=1}^{s} |Aw^{(t)}|^2 \\ &= |Av^{(1)}|^2 + \sum_{t=1}^{s} |Aw^{(t)}|^2 - |Av^{(1)}|^2 \\ &\leq \max_{u^{(1)}, u^{(2)} \ldots u^{(s+1)} \text{ orthonormal}} \sum_{t=1}^{s+1} |Au^{(t)}|^2 - |Av^{(1)}|^2 \\ &= \sigma_1(A)^2 + \sigma_2(A)^2 + \ldots \sigma_{s+1}(A)^2 - |Av^{(1)}|^2, \qquad (9.9)\end{aligned}$$

where we have applied the fact that for any k, the k-dimensional SVD subspace maximizes the sum of squared projections among all subspaces of dimension at most k.

Now, we use the inductive assumption on A' with the orthonormal basis $v^{(2)}, v^{(3)}, \ldots v^{(r)}$. This yields for $t, 2 \leq t \leq r$,

$$\begin{aligned}&\sum_{s=t+1}^{r} ({v^{(t)}}^T A'^T A' v^{(s)})^2 \\ &\leq |A'v^{(t)}|^2 (\sigma_1^2(A') + \sigma_2^2(A') + \ldots \sigma_{t-1}^2(A') - |A'v^{(2)}|^2 - |A'v^{(3)}|^2 - \ldots |A'v^{(t)}|^2)\end{aligned}$$

Note that for $t \geq 2$, we have $A'v^{(t)} = Av^{(t)}$. So, we get using (9.9)

$$\begin{aligned}&\sum_{s=t+1}^{r} ({v^{(t)}}^T A^T A v^{(s)})^2 \\ &\leq |Av^{(t)}|^2 (\sigma_1^2(A) + \sigma_2^2(A) + \ldots \sigma_t^2(A) - |Av^{(1)}|^2 - |Av^{(2)}|^2 - \ldots |Av^{(t)}|^2).\end{aligned}$$

This together with (9.8) finishes the proof of the Theorem. □

We will use Theorem (9.8) to prove Theorem (9.9) below. Theorem (9.9) says that we can get good "left projections" from "good right projections". One important difference from the exact case is that now we have to be more careful of "near singularities", i.e. the upper bounds in the Theorem (9.9) will depend on a term

$$\sum_{t=1}^{k} \frac{1}{|Av^{(t)}|^2}.$$

If some of the $|Av^{(t)}|$ are close to zero, this term is large and the bounds can become useless. This is not just a technical problem. In defining $u^{(t)}$ in Theorem (9.7) as $Av^{(t)}/|Av^{(t)}|$, the hypotheses exclude t for which the denominator is zero. Now since we are dealing with approximations, it is not only the zero denominators that bother us, but also small denominators. We will have to exclude these too (as in Corollary (9.10) below) to get a reasonable bound.

Theorem 9.9. *Suppose A is a matrix and $v^{(1)}, \ldots v^{(k)}$ are orthonormal and let $\Delta = \Delta(A, v^{(1)}, v^{(2)}, \ldots v^{(k)})$ be as in (9.7). Let*

$$u^{(t)} = \frac{Av^{(t)}}{|Av^{(t)}|} \quad \textit{for } t = 1, 2, \ldots k.$$

Then

$$\begin{aligned} \|\sum_{t=1}^{k} u^{(t)} u^{(t)^T} A - A\|_F^2 &\leq \|A - \sum_{t=1}^{k} Av^{(t)} v^{(t)^T}\|_F^2 \\ &\quad + \left(\sum_{t=1}^{k} \frac{2}{|Av^{(t)}|^2}\right)\left(\sum_{t=1}^{k} |Av^{(t)}|^2\right)\Delta \end{aligned}$$

$$\begin{aligned} \|\sum_{t=1}^{k} u^{(t)} u^{(t)^T} A - A\|_2^2 &\leq \|A - \sum_{t=1}^{k} Av^{(t)} v^{(t)^T}\|_2^2 \\ &\quad + \left(\sum_{t=1}^{k} \frac{2}{|Av^{(t)}|^2}\right)\left(\sum_{t=1}^{k} |Av^{(t)}|^2\right)\Delta. \end{aligned}$$

Proof. Complete $\{v^{(1)}, v^{(2)}, \ldots v^{(k)}\}$ to an orthonormal set $\{v^{(1)}, v^{(2)}, \ldots v^{(r)}\}$ such that $\{Av^{(t)} : t = 1, 2, \ldots r\}$ span the range of A. Let

$$w^{(t)^T} = v^{(t)^T} A^T A - |Av^{(t)}|^2 v^{(t)^T}$$

be the component of $v^{(t)^T} A^T A$ orthogonal to $v^{(t)^T}$. We have

$$u^{(t)} u^{(t)^T} A = \frac{Av^{(t)} v^{(t)^T} A^T A}{|Av^{(t)}|^2} = Av^{(t)} v^{(t)^T} + Av^{(t)} w^{(t)^T}.$$

Using $||X+Y||_F^2 = \mathrm{Tr}((X^T+Y^T)(X+Y)) = ||X||_F^2 + ||Y||_F^2 + 2\mathrm{Tr}X^TY$ and the convention that t runs over $1, 2, \ldots k$, we have

$$
\begin{aligned}
||\sum_t u^{(t)}u^{(t)^T}A - A||_F^2 &= \left|\left|\sum_t Av^{(t)}v^{(t)^T} + \sum_t \frac{Av^{(t)}w^{(t)^T}}{|Av^{(t)}|^2} - A\right|\right|_F^2 \\
= \quad & ||A - \sum_t Av^{(t)}v^{(t)^T}||_F^2 + \left(\sum_t \left|\frac{Av^{(t)}}{|Av^{(t)}|^2}\right|\left|w^{(t)}\right|\right)^2 \\
& -2\sum_{s=1}^{r}\sum_t (v^{(s)^T}w^{(t)})\frac{v^{(t)^T}A^T}{|Av^{(t)}|^2}(A - \sum_t Av^{(t)}v^{(t)^T})v^{(s)} \\
\leq \quad & ||A - \sum_t Av^{(t)}v^{(t)^T}||_F^2 + \left(\sum_t |w^{(t)}|^2\right)\left(\sum_t \frac{1}{|Av^{(t)}|^2}\right) - 2\sum_{s=k+1}^{r}\sum_t \frac{(v^{(t)^T}A^TAv^{(s)})^2}{|Av^{(t)}|^2} \\
& \text{since } (A - \textstyle\sum_t Av^{(t)}v^{(t)^T})v^{(s)} = 0 \quad \text{for } s \leq k \text{ and } v^{(s)^T}w^{(t)} = v^{(s)^T}A^TAv^{(t)} \\
\leq \quad & ||A - \sum_t Av^{(t)}v^{(t)^T}||_F^2 + \left(\sum_t \frac{1}{|Av^{(t)}|^2}\right)\left(2\sum_t \sum_{s=t+1}^{r} (v^{(t)^T}A^TAv^{(s)})^2\right) \\
\leq \quad & |A - \sum_t Av^{(t)}v^{(t)^T}||_F^2 + \left(\sum_t \frac{2}{|Av^{(t)}|^2}\right)\left(\sum_t |Av^{(t)}|^2\right)\Delta,
\end{aligned}
$$

using Theorem (9.8).

For the 2-norm, the argument is similar. Suppose a vector p achieves

$$||\sum_t u^{(t)}u^{(t)^T}A - A||_2 = |(\sum_t u^{(t)}u^{(t)^T}A - A)p|.$$

We now use

$$|(X+Y)p|^2 = p^TX^TXp + p^TY^TYp + 2p^TX^TYp$$

to get

$$
\begin{aligned}
&||\sum_t u^{(t)}u^{(t)^T}A - A||_2^2 \leq ||A - \sum_t Av^{(t)}v^{(t)^T}||_2^2 \\
&+ \left(\sum_t |w^{(t)}|^2\right)\left(\sum_t \frac{1}{|Av^{(t)}|^2}\right) - 2\sum_t (p^Tw^{(t)})\frac{v^{(t)^T}A^T}{|Av^{(t)}|^2}(A - \sum_t Av^{(t)}v^{(t)^T})p.
\end{aligned}
$$

If now we write $p = p^{(1)} + p^{(2)}$, where $p^{(1)}$ is the component of p in the span of $v^{(1)}, v^{(2)}, \ldots v^{(k)}$, then we have

$$
\begin{aligned}
\sum_t (p^Tw^{(t)})\frac{v^{(t)^T}A^T}{|Av^{(t)}|^2}(A - \sum_t Av^{(t)}v^{(t)^T})p &= \sum_t (p^{(2)^T}w^{(t)})\frac{v^{(t)^T}A^T}{|Av^{(t)}|^2}Ap^{(2)} \\
&= \frac{\sum_t (v^{(t)^T}A^TAp^{(2)})^2}{|Av^{(t)}|^2},
\end{aligned}
$$

where we have used the fact that $p^{(2)}$ is orthogonal to $v^{(t)}$ to get ${p^{(2)}}^T w^{(t)} = {v^{(t)}}^T A^T A p^{(2)}$.

□

We will apply the Theorem as follows. As remarked earlier, we have to be careful about near singularities. Thus while we seek a good approximation of rank k or less, we cannot automatically take all of the k terms. Indeed, we only take terms for which $|Av^{(t)}|$ is at least a certain threshold.

Corollary 9.10. *Suppose A is a matrix, δ a positive real and $v^{(1)}, \ldots v^{(k)}$ are orthonormal vectors produced by a randomized algorithm and suppose*

$$\mathsf{E}\left(\sum_{j=1}^{t}\left(\sigma_j^2(A) - |Av^{(j)}|^2\right)\right) \leq \delta||A||_F^2 \quad t = 1, 2, \ldots k.$$

Let

$$u^{(t)} = \frac{Av^{(t)}}{|Av^{(t)}|} \quad \textit{for } t = 1, 2, \ldots k.$$

Define l to be the largest integer in $\{1, 2, \ldots k\}$ such that $|Av^{(l)}|^2 \geq \sqrt{\delta}||A||_F^2$. Then,

$$\mathsf{E}\,||A - \sum_{t=1}^{l} u^{(t)}{u^{(t)}}^T A||_F^2 \leq \mathsf{E}\,||A - A\sum_{t=1}^{k} v^{(t)}{v^{(t)}}^T||_F^2 + 3k\sqrt{\delta}||A||_F^2.$$

$$\mathsf{E}\,||A - \sum_{t=1}^{l} u^{(t)}{u^{(t)}}^T A||_2^2 \leq \mathsf{E}\,||A - A\sum_{t=1}^{k} v^{(t)}{v^{(t)}}^T||_2^2 + 3k\sqrt{\delta}||A||_F^2$$

Proof. We apply the Theorem with k replaced by l and taking expectations of both sides (which are now random variables) to get

$$\begin{aligned}
&\mathsf{E}\,||A - \sum_{t=1}^{l} u^{(t)}{u^{(t)}}^T||_F^2 \leq \mathsf{E}\,||A - A\sum_{t=1}^{l} v^{(t)}{v^{(t)}}^T||_F^2 + \\
&+\frac{2k}{\sqrt{\delta}}\mathsf{E}\left(\sum_{t=1}^{l}\left(\sigma_t^2(A) - |Av^{(t)}|^2\right)\right) \\
\leq\ &\mathsf{E}\,||A - A\sum_{t=1}^{k} v^{(t)}{v^{(t)}}^T||_F^2 + \sum_{t=l+1}^{k}|Av^{(t)}|^2 + 2k\sqrt{\delta}||A||_F^2,
\end{aligned}$$

where, we have used the fact that from the minimax principle and $|Av^{(1)}| \geq |Av^{(2)}| \geq \ldots |Av^{(k)}| > 0$, we get that $\sigma_t(A) \geq |Av^{(t)}|$ for $t = 1, 2, \ldots k$. Now first assertion in the Corollary follows. For the 2-norm bound, the proof is similar. Now we use the fact that

$$||A - A\sum_{t=1}^{l} v^{(t)}{v^{(t)}}^T||_2^2 \leq ||A - A\sum_{t=1}^{k} v^{(t)}{v^{(t)}}^T||_2^2 + \sum_{t=l+1}^{k}|Av^{(t)}|^2.$$

To see this, if p is the top left singular vector of $A - A\sum_{t=1}^{l} v^{(t)}v^{(t)^T}$, then

$$\begin{aligned} |p^T(A - A\sum_{t=1}^{l} v^{(t)}v^{(t)^T})|^2 &= p^T AA^T p - p^T A\sum_{t=1}^{l} v^{(t)}v^{(t)^T}A^T p \\ &\leq ||A - A\sum_{t=1}^{k} v^{(t)}v^{(t)^T}||_2^2 + \sum_{t=l+1}^{k} |p^T A v^{(t)}|^2. \end{aligned}$$

□

9.5 SVD by sampling rows and columns

Suppose A is an $m \times n$ matrix and $\epsilon > 0$ and c a real number in $[0, 1]$. In this section, we will use several constants which we denote $c_1, c_2 \ldots$ which we do not specify.

We pick a sample of

$$s = \frac{c_1 k^5}{c\epsilon^4}$$

columns of A according to $\mathrm{LS}_{\mathrm{col}}(A, c)$ and scale to form an $m \times s$ matrix C. Then we sample a set of s rows of C according to a $\mathrm{LS}_{\mathrm{row}}(C, c)$ distribution to form a $s \times s$ matrix W. By Theorem 9.2, we have

$$\mathsf{E}\, ||C^T C - W^T W||_F \leq \frac{1}{\sqrt{cs}} \mathsf{E}\, ||C||_F^2 = \frac{c_2\epsilon^2}{k^{2.5}}||A||_F^2, \tag{9.10}$$

where we have used Hölder's inequality ($\mathsf{E}\, X \leq (\mathsf{E}\, X^2)^{1/2}$) and the fact that $\mathsf{E}\, ||C||_F^2 = \mathsf{E}\, \mathrm{Tr}(CC^T) = \mathrm{Tr}(AA^T)$.

We now find the SVD of $W^T W$, (an $s \times s$ matrix!) say

$$W^T W = \sum_t \sigma_t^2(W) v^{(t)} v^{(t)^T}.$$

We first wish to claim that $\sum_{t=1}^{k} v^{(t)}v^{(t)^T}$ forms a "good right projection" for C. This follows from Lemma (9.3) with C replacing A and W replacing C in that Lemma and right projections instead of left projections. Hence we get (using (9.10))

$$\mathsf{E}\, ||C - C\sum_{t=1}^{k} v^{(t)}v^{(t)^T}||_F^2 \leq \mathsf{E}\, ||C||_F^2 - \mathsf{E}\, \sum_{t=1}^{k} \sigma_t^2(C) + \frac{c_3\epsilon^2}{k^2}||A||_F^2 \tag{9.11}$$

$$\begin{aligned} \mathsf{E}\, ||C - C\sum_{t=1}^{k} v^{(t)}v^{(t)^T}||_2^2 &\leq \mathsf{E}\, \sigma_{k+1}(C)^2 + (2+4k)O(\frac{\epsilon^2}{k^3})\mathsf{E}\, ||C||_F^2 \quad (9.12) \\ &\leq \sigma_{k+1}^2(A) + \frac{c_4\epsilon^2}{k^2}||A||_F^2. \quad (9.13) \end{aligned}$$

Since $||C - C\sum_{t=1}^{k} v^{(t)}v^{(t)^T}||_F^2 = ||C||_F^2 - \sum_{t=1}^{k} |Cv^{(t)}|^2$, we get from (9.13)

$$\mathsf{E} \sum_{t=1}^{k} \left(\sigma_t^2(C) - |Cv^{(t)}|^2\right) \leq \frac{c_5\epsilon^2}{k^2}||A||_F^2. \tag{9.14}$$

(9.13) also yields

$$\begin{aligned} \mathsf{E}\, ||C - C\sum_{t=1}^{k} v^{(t)}v^{(t)^T}||_F^2 &\leq ||A||_F^2 - \sum_{t=1}^{k} \sigma_t^2(A) + ||A||_F^2 \frac{c_6\epsilon^2}{k^2} \\ \text{Thus,}\quad \mathsf{E}\, ||C - C\sum_{t=1}^{k} v^{(t)}v^{(t)^T}||_F^2 &\leq \sum_{t=k+1}^{n} \sigma_t^2(A) + \frac{c_6\epsilon^2}{k^2}||A||_F^2. \end{aligned} \tag{9.15}$$

Now we wish to use Corollary (9.10) to derive a good left projection for C from the right projection above. To this end, we define

$$u^{(t)} = \frac{Cv^{(T)}}{|Cv^{(t)}|} \quad \text{for } t = 1, 2, \ldots k.$$

Define l to be the largest integer in $\{1, 2, \ldots k\}$ such that $|Cv^{(l)}|^2 \geq \frac{\sqrt{c_5}\epsilon}{k}||A||_F^2$. Then from the Corollary, we get

$$\begin{aligned} \mathsf{E}\, ||C - \sum_{t=1}^{l} u^{(t)}u^{(t)^T}C||_F^2 &\leq \mathsf{E}\, ||C - C\sum_{t=1}^{k} v^{(t)}v^{(t)^T}||_F^2 + O(\epsilon)||A||_F^2 \\ &\leq \sum_{t=k+1}^{n} \sigma_t^2(A) + O(\epsilon)||A||_F^2. \end{aligned} \tag{9.16}$$

$$\mathsf{E}\, ||C - \sum_{t=1}^{l} u^{(t)}u^{(t)^T}C||_2^2 \leq \sigma_{k+1}^2(A) + O(\epsilon)||A||_F^2. \tag{9.17}$$

Finally,we use Lemma (9.5) to argue that $\sum_{t=1}^{l} u^{(t)}u^{(t)^T}$ is a good left projection for A. To do so, we first note that $||\sum_{t=1}^{l} u^{(t)}u^{(t)^T}||_F \leq \sum_{t=1}^{l} |u^{(t)}|^2 \leq k$. So,

$$\begin{aligned} \mathsf{E}\, ||A - \sum_{t=1}^{l} u^{(t)}u^{(t)^T}A||_F^2 &\leq \mathsf{E}\, ||C - \sum_{t=1}^{l} u^{(t)}u^{(t)^T}C||_F^2 + \frac{1}{\sqrt{cs}}||A||_F^2 k(2+k) \\ &\leq \sum_{t=k+1}^{n} \sigma_t^2(A) + O(\epsilon)||A||_F^2 \\ \mathsf{E}\, ||A - \sum_{t=1}^{l} u^{(t)}u^{(t)^T}A||_2^2 &\leq \sigma_{k+1}^2(A) + O(\epsilon)||A||_F^2. \end{aligned}$$

Thus, we get the following lemma:

Lemma 9.11. *Suppose we are given an $m \times n$ matrix A, a positive integer $k \leq m, n$ and a real $\epsilon > 0$. Then for the $u^{(1)}, u^{(2)}, \ldots u^{(l)}$ produced by the constant-time-SVD algorithm, we have the following two bounds:*

$$\mathsf{E}\ ||A - \sum_{t=1}^{l} u^{(t)} u^{(t)^T} A||_F^2 \leq \sum_{t=k+1}^{n} \sigma_t^2(A) + \epsilon ||A||_F^2$$

$$\mathsf{E}\ ||A - \sum_{t=1}^{l} u^{(t)} u^{(t)^T} A||_2^2 \leq \sigma_{k+1}^2(A) + \epsilon ||A||_F^2.$$

The proof is already given.

Algorithm: Constant-time SVD

1. Pick a sample of
$$s = \frac{c_8 k^5}{c\epsilon^4}$$
columns of A according to $\mathrm{LS}_{\mathtt{col}}(A, c)$ and scale to form an $m \times s$ matrix C.

2. Sample a set of s rows of C according to a $\mathrm{LS}_{\mathtt{row}}(C, c)$ distribution and scale to form a $s \times s$ matrix W.

3. Find the SVD of $W^T W$:
$$W^T W = \sum_t \sigma_t^2(W) v^{(t)} v^{(t)^T}.$$

4. Compute
$$u^{(t)} = \frac{C v^{(t)}}{|C v^{(t)}|} \quad \text{for } t = 1, 2, \ldots k.$$
Let l to be the largest integer in $\{1, 2, \ldots k\}$ such that
$$|C v^{(l)}|^2 \geq c_9 \epsilon ||C||_F^2 / k.$$

5. Return
$$\sum_{t=1}^{l} u^{(t)} u^{(t)^T} A$$
as the approximation to A.

9.6 CUR: An interpolative low-rank approximation

In this section, we wish to describe an algorithm to get an approximation of any matrix A given just a sample of rows and a sample of columns of A. Clearly if the sample is picked according to the uniform distribution, this attempt would fail in general. We will see that again the length squared distribution comes to our rescue; indeed, we will show that if the samples are picked according to the length squared or approximate length squared distributions, we can get an approximation for A. Again, this will hold for an arbitrary matrix A.

First suppose A is a $m \times n$ matrix and R (R for rows) is a $s \times n$ matrix construced by picking s rows of A in i.i.d. samples, each according to $\mathrm{LS}_{\mathrm{row}(A,c)}$ and scaled. Similarly, let C (for columns) be a $m \times s$ matrix consisting of columns picked according to $\mathrm{LS}_{\mathrm{col}(A,c)}$ and scaled. The motivating question for this section is: Can we get an approximation to A given just C, R?

Intuitively, this should be possible since we know that $CC^T \approx AA^T$ and $R^TR \approx A^TA$. Now it is easy to see that if we are given both AA^T and A^TA and A is in "general position", i.e., say all its singular values are distinct, then A can be found: indeed, if the SVD of A is

$$A = \sum_t \sigma_t(A) u^{(t)} v^{(t)^T},$$

then

$$AA^T = \sum_t \sigma_t^2(A) u^{(t)} u^{(t)^T} \qquad A^TA = \sum_t \sigma_t^2(A) v^{(t)} v^{(t)^T},$$

and so from the SVD's of AA^T, A^TA, the SVD of A can be read off if the $\sigma_t(A)$ are all distinct. [This is not the case if the σ_t are not distinct; for example, for any square A with orthonormal columns, $AA^T = A^TA = I$.] The above idea leads intuitively to the guess that at least in general position, C, R are sufficient to produce some approximation to A.

The approximation of A by the product CUR is reminiscent of the usual PCA approximation based on taking the leading k terms of the SVD decomposition. There, instead of C, R, we would have orthonormal matrices consisting of the leading singular vectors and instead of U, the diagonal matrix of singular values. The PCA decomposition of course gives the best rank-k approximation, whereas what we will show below for CUR is only that its error is bounded in terms of the best error we can achieve. There are two main advantages of CUR over PCA:

1. CUR can be computed much faster from A and also we only need to make two passes over A which can be assumed to be stored on external memory.

2. CUR preserves the sparsity of A - namely C, R are columns and rows of A itself. (U is a small matrix since typically s is much smaller than m, n). So any further matrix vector products Ax can be approximately computed as $C(U(Rx))$ quickly.

The main theorem of this section is the following.

Theorem 9.12. *Suppose A is any $m \times n$ matrix, C is any $m \times s$ matrix of rank at least k. Suppose $i_1, i_2, \ldots i_s$ are obtained from s i.i.d. trials each according to probabilities $\{p_1, p_2, \ldots p_m\}$ conforming to $LS_{rows(A,c)}$ and let R be the $s \times n$ matrix with t th row equal to $A_{i_t}/\sqrt{sp_{i_t}}$. Then, from $C, R, \{i_t\}$, we can find an $s \times s$ matrix U such that*

$$\begin{aligned} \mathsf{E}\left(\|CUR - A\|_F\right) &\leq \|A - A_k\|_F + \sqrt{\frac{k}{cs}}||A||_F + \sqrt{2}k^{\frac{1}{4}}||AA^T - CC^T||_F^{1/2} \\ \mathsf{E}\left(\|CUR - A\|_2\right) &\leq \|A - A_k\|_2 + \sqrt{\frac{k}{cs}}||A||_F + \sqrt{2}||AA^T - CC^T||_F^{1/2} \end{aligned}$$

Proof. The selection of rows and scaling used to obtain R from A can be represented by as

$$R = DA,$$

where D has only one non-zero entry per row. Let the SVD of C be

$$C = \sum_{t=1}^{r} \sigma_t(C) x^{(t)} y^{(t)^T}.$$

By assumption $\sigma_k(C) > 0$. Then the SVD of $C^T C$ is

$$C^T C = \sum_{t=1}^{r} \sigma_t^2(C) y^{(t)} y^{(t)^T}.$$

Then, we prove the theorem with U defined by

$$U = \sum_{t=1}^{k} \frac{1}{\sigma_t^2(C)} y^{(t)} y^{(t)^T} C^T D^T.$$

Then, using the orthonormality of $\{x^{(t)}\}, \{y^{(t)}\}$,

$$\begin{aligned} CUR &= \sum_{t=1}^{r} \sigma_t(C) x^{(t)} y^{(t)^T} \sum_{s=1}^{k} \frac{1}{\sigma_s^2(C)} y^{(s)} y^{(s)^T} \sum_{p=1}^{r} \sigma_p(C) y^{(p)} x^{(p)^T} D^T DA \\ &= \sum_{t=1}^{k} x^{(t)} x^{(t)^T} D^T DA \end{aligned}$$

Consider the matrix multiplication

$$\left(\sum_{t=1}^{k} x^{(t)} x^{(t)^T}\right)(A).$$

$D^T D$ above can be viewed precisely as selecting some rows of the matrix A and the corresponding columns of $\sum_t x^{(t)} x^{(t)^T}$ with suitable scaling. Applying

Theorem 9.1 directly, we thus get using $||\sum_{t=1}^{k} x^{(t)}x^{(t)^T}||_F^2 = k$. In the theorem, one is selecting columns of the first matrix according to $\mathrm{LS}_{\mathrm{col}}$ of that matrix; here symmetrically, we are selecting rows of the second matrix according to $\mathrm{LS}_{\mathrm{row}}$ of that matrix.

$$\mathsf{E}\left|\left|\sum_{t=1}^{k} x^{(t)}x^{(t)^T}D^TDA - \sum_{t=1}^{k} x^{(t)}x^{(t)^T}A\right|\right|_F^2 \leq \frac{k}{cs}||A||_F^2.$$

Thus,

$$\mathsf{E}\,||CUR - \sum_{t=1}^{k} x^{(t)}x^{(t)^T}A||_F^2 \leq \frac{k}{cs}||A||_F^2.$$

Next, from Lemma 9.3 it follows that

$$\begin{aligned}
\|\sum_{t=1}^{k} x^{(t)}x^{(t)^T}A - A\|_F^2 &\leq \|A - A_k\|_F^2 + 2\sqrt{k}\|AA^T - CC^T\|_F \\
\|\sum_{t=1}^{k} x^{(t)}x^{(t)^T}A - A\|_2^2 &\leq \|A - A_k\|_2 + 2\|AA^T - CC^T\|_F.
\end{aligned}$$

Now the theorem follows using the triangle inequality on the norms. □

As a corollary, we have the following:

Corollary 9.13. *Suppose we are given C, a set of independently chosen columns of A from $LS_{col(A,c)}$ and R, a set of s independently chosen rows of A from $LS_{rows(A,c)}$. Then, in time $O((m+n)s^2)$, we can find an $s \times s$ matrix U such that for any k,*

$$\mathsf{E}\ (\|A - CUR\|_F) \leq \|A - A_k\|_F + \left(\frac{k}{s}\right)^{1/2}\|A\|_F + \left(\frac{4k}{s}\right)^{1/4}\|A\|_F$$

The following open problem, if answered affirmatively, would generalize the theorem.

Problem Suppose A is any $m \times n$ matrix and C, R are **any** $m \times s$ and $s \times n$ (respectively) matrices with

$$||AA^T - CC^T||_F, ||A^TA - R^TR||_F \leq \delta||A||_F^2.$$

Then, from just C, R, can we find a $s \times s$ matrix U such that

$$||A - CUR||_F \leq \mathrm{poly}(\frac{\delta}{s})||A||_F?$$

So we do not assume that R is a random sample as in the theorem.

9.7 Discussion

Sampling from the length square distribution was introduced in a paper by Frieze, Kannan and Vempala [FKV98, FKV04] in the context of a constant-time algorithm for low-rank approximation. It has been used many times subsequently. There are several advantages of sampling-based algorithms for matrix approximation. The first is efficiency. The second is the nature of the approximation, namely it is often interpolative, i.e., uses rows/columns of the original matrix. Finally, the methods can be used in the streaming model where memory is limited and entries of the matrix arrive in arbitrary order.

The analysis for matrix multiplication is originally due to Drineas and Kannan [DK01]. The linear-time low-rank approximation was given by Drineas et al. [DFK$^+$04]. The CUR decomposition first appeared in [DK03]. The best-know sample complexity for the constant-time algorithm is $O(k^2/\epsilon^4)$ and other refinements are given in [DKM06a, DKM06b, DKM06c]. An alternative sampling method which sparsifies a given matrix and uses a low-rank approximation of the sparse matrix was given in [AM07].

We conclude this section with a description of some typical applications. A recommendation system is a marketing tool with wide use. Central to this is the consumer-product matrix A where A_{ij} is the "utility" or "preference" of consumer i for product j. If the entire matrix were available, the task of the system is simple - whenever a user comes up, it just recommends to the user the product(s) of maximum utility to the user. But this assumption is unrealistic; market surveys are costly, especially if one wants to ask each consumer. So, the essential problem in Recommendation Systems is Matrix Reconstruction - given only a sampled part of A, reconstruct (implicitly, because writing down the whole of A requires too much space) an approximation A' to A and make recommendations based on A'. A natural assumption is to say that we have a set of sampled rows (we know the utilities of some consumers- at least their top choices) and a set of sampled columns (we know the top buyers of some products). This model very directly suggests the use of the CUR decomposition below which says that for any matrix A given a set of sampled rows and columns, we can construct an approximation A' to A from them. Some well-known recommendation systems in practical use relate to on-line book sellers, movie renters etc.

In the first mathematical model for Recommendation Systems Azar et al. [AFKM01] assumed a generative model where there were k types of consumers and each is a draw from a probability distribution (a mixture model). It is easy to see then that A is close to a low-rank matrix. The CUR type model and analysis using CUR decomposition was by [DKR02].

We note an important philosophical difference in the use of sampling here from previous topics discussed. Earlier, we assumed that there was a huge matrix A explicitly written down somewhere and since it was too expensive to compute with all of it, one used sampling to extract a part of it and computed with this. Here, the point is that it is expensive to get the whole of A, so we have to do with a sample from which we "reconstruct" implicitly the whole.

Bibliography

[ADHP09] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat, *Np-hardness of euclidean sum-of-squares clustering*, Machine Learning **75** (2009), no. 2, 245–248.

[AFKM01] Y. Azar, A. Fiat, A. Karlin, and F. McSherry, *Spectral analysis of data*, Proc. of STOC, 2001, pp. 619–626.

[AGH+15] Anima Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky, *Tensor decompositions for learning latent variable models (a survey for alt)*, International Conference on Algorithmic Learning Theory, Springer, 2015, pp. 19–38.

[AK05] S. Arora and R. Kannan, *Learning mixtures of arbitrary gaussians*, Annals of Applied Probability **15** (2005), no. 1A, 69–92.

[AKS98] N. Alon, M. Krivelevich, and B. Sudakov, *Finding a large hidden clique in a random graph*, Random Structures and Algorithms **13** (1998), 457–466.

[AM05] D. Achlioptas and F. McSherry, *On spectral learning of mixtures of distributions*, Proc. of COLT, 2005.

[AM07] Dimitris Achlioptas and Frank McSherry, *Fast computation of low-rank matrix approximations*, J. ACM **54** (2007), no. 2.

[ARV04] Sanjeev Arora, Satish Rao, and Umesh Vazirani, *Expander flows, geometric embeddings and graph partitioning*, STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing, 2004, pp. 222–231.

[AS12] Pranjal Awasthi and Or Sheffet, *Improved spectral-norm bounds for clustering*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings, 2012, pp. 37–49.

[Bha97] R. Bhatia, *Matrix analysis*, Springer, 1997.

[Bop87] R. Boppana, *Eigenvalues and graph bisection: An average-case analysis*, Proceedings of the 28th IEEE Symposium on Foundations of Computer Science (1987), 280–285.

[Bru09] S. C. Brubaker, *Robust pca and clustering on noisy mixtures*, Proc. of SODA, 2009.

[BV08] S. C. Brubaker and S. Vempala, *Isotropic pca and affine-invariant clustering*, Building Bridges Between Mathematics and Computer Science (M. Grötschel and G. Katona, eds.), Bolyai Society Mathematical Studies, vol. 19, 2008.

[CKVW06] David Cheng, Ravi Kannan, Santosh Vempala, and Grant Wang, *A divide-and-merge methodology for clustering*, ACM Trans. Database Syst. **31** (2006), no. 4, 1499–1525.

[CR08a] K. Chaudhuri and S. Rao, *Beyond gaussians: Spectral methods for learning mixtures of heavy-tailed distributions*, Proc. of COLT, 2008.

[CR08b] ———, *Learning mixtures of product distributions using correlations and independence*, Proc. of COLT, 2008.

[Das99] S. DasGupta, *Learning mixtures of gaussians*, Proc. of FOCS, 1999.

[DFK+04] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay, *Clustering large graphs via the singular value decomposition*, Machine Learning **56** (2004), 9–33.

[DHKM07] Anirban Dasgupta, John Hopcroft, Ravi Kannan, and Pradipta Mitra, *Spectral clustering with limited independence*, Proc. of SODA (Philadelphia, PA, USA), Society for Industrial and Applied Mathematics, 2007, pp. 1036–1045.

[DHKS05] A. Dasgupta, J. Hopcroft, J. Kleinberg, and M. Sandler, *On learning mixtures of heavy-tailed distributions*, Proc. of FOCS, 2005.

[DHS01] R. O. Duda, P.E. Hart, and D.G. Stork, *Pattern classification*, John Wiley & Sons, 2001.

[DK01] Petros Drineas and Ravi Kannan, *Fast monte-carlo algorithms for approximate matrix multiplication*, In Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, 2001, pp. 452–459.

[DK03] Petros Drineas and Ravi Kannan, *Pass efficient algorithms for approximating large matrices*, SODA '03: Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, 2003, pp. 223–232.

[DKM06a] P. Drineas, R. Kannan, and M. Mahoney, *Fast monte carlo algorithms for matrices I: Approximating matrix multiplication*, SIAM J. on Computing **36** (2006), 132–157.

[DKM06b] ———, *Fast monte carlo algorithms for matrices II: Computing a low-rank approximation to a matrix*, SIAM J. on Computing **36** (2006), 158–183.

[DKM06c] ———, *Fast monte carlo algorithms for matrices III: Computing a compressing approximate matrix decomposition*, SIAM J. on Computing **36** (2006), 184–206.

[DKR02] P. Drineas, I. Kerenidis, and P. Raghavan, *Competitive Recommendation Systems*, Proceedings of the 34th Annual ACM Symposium on Theory of Computing (2002), 82–90.

[DS00] S. DasGupta and L. Schulman, *A two-round variant of em for gaussian mixtures*, Proc. of UAI, 2000.

[FK81] Z. Füredi and J. Komlós, *The eigenvalues of random symmetric matrices*, Combinatorica **1** (1981), no. 3, 233–241.

[FK99] A. Frieze and R. Kannan, *Quick approximation to matrices and applications*, Combinatorica **19** (1999), no. 2, 175–200.

[FKV98] A. Frieze, R. Kannan, and S. Vempala, *Fast monte-carlo algorithms for finding low-rank approximations*, Proc. of FOCS, 1998, pp. 370–378.

[FKV04] Alan Frieze, Ravi Kannan, and Santosh Vempala, *Fast monte-carlo algorithms for finding low-rank approximations*, J. ACM **51** (2004), no. 6, 1025–1041.

[FSO06] J. Feldman, R. A. Servedio, and R. O'Donnell, *Pac learning axis-aligned mixtures of gaussians with no separation assumption*, Proc. of COLT, 2006, pp. 20–34.

[Fuk90] K. Fukunaga, *Introduction to statistical pattern recognition*, Academic Press, 1990.

[GJ79] M. R. Garey and David S. Johnson, *Computers and intractability: A guide to the theory of np-completeness*, W. H. Freeman, 1979.

[GvL96] G. H. Golub and C. F. van Loan, *Matrix computations*, 3rd ed., Johns Hopkins University Press, 1996.

[HK13] Daniel Hsu and Sham M Kakade, *Learning mixtures of spherical gaussians: moment methods and spectral decompositions*, Proceedings of the 4th conference on Innovations in Theoretical Computer Science, 2013, pp. 11–20.

[HPV02] Sariel Har-Peled and Kasturi R. Varadarajan, *Projective clustering in high dimensions using core-sets*, Symposium on Computational Geometry, 2002, pp. 312–318.

[Kel06] Jonathan A. Kelner, *Spectral partitioning, eigenvalue bounds, and circle packings for graphs of bounded genus*, SIAM J. Comput. **35** (2006), no. 4, 882–902.

[KK10] Amit Kumar and Ravindran Kannan, *Clustering with spectral norm and the k-means algorithm*, 51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA, 2010, pp. 299–308.

[KMN+04] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu, *A local search approximation algorithm for k-means clustering*, Comput. Geom. **28** (2004), no. 2-3, 89–112.

[KMV10] Adam Kalai, Ankur Moitra, and Gregory Valiant, *Efficiently learning mixtures of two gaussians*, 06 2010, pp. 553–562.

[KSV08] R. Kannan, H. Salmasian, and S. Vempala, *The spectral method for general mixture models*, SIAM Journal on Computing **38** (2008), no. 3, 1141–1156.

[KVV04] Ravi Kannan, Santosh Vempala, and Adrian Vetta, *On clusterings: Good, bad and spectral*, J. ACM **51** (2004), no. 3, 497–515.

[LP86] F. Lust-Piquard, *Inégalites de khinchin dans c_p $(1 < p < \infty)$*, C.R. Acad. Sci., Paris **303** (1986), 289–292.

[LR99] F. T. Leighton and S. Rao, *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms*, J. ACM **46** (1999), no. 6, 787–832.

[LV07] L. Lovász and S. Vempala, *The geometry of logconcave functions and sampling algorithms*, Random Structures and Algorithms **30** (2007), no. 3, 307–358.

[McS01] F. McSherry, *Spectral partitioning of random graphs*, FOCS, 2001, pp. 529–537.

[MT82] N. Megiddo and A. Tamir, *On the complexity of locating facilities in the plane*, Operations Research Letters **I** (1982), 194–197.

[MV10] Ankur Moitra and Gregory Valiant, *Settling the polynomial learnability of mixtures of gaussians*, 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, IEEE, 2010, pp. 93–102.

[PRTV98] C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, *Latent semantic indexing: A probabilistic analysis*, Proc. of PODS, 1998.

[Rud99] M. Rudelson, *Random vectors in the isotropic position*, Journal of Functional Analysis **164** (1999), 60–72.

[SJ89] A. Sinclair and M. Jerrum, *Approximate counting, uniform generation and rapidly mixing markov chains*, Information and Computation **82** (1989), 93–133.

[ST07] Daniel A. Spielman and Shang-Hua Teng, *Spectral partitioning works: Planar graphs and finite element meshes*, Linear Algebra and its Applications **421** (2007), no. 2-3, 284 – 305.

[Str88] Gilbert Strang, *Linear algebra and its applications*, Brooks Cole, 1988.

[Vu05] V. H. Vu, *Spectral norm of random matrices*, Proc. of STOC, 2005, pp. 423–430.

[VW04] S. Vempala and G. Wang, *A spectral algorithm for learning mixtures of distributions*, Journal of Computer and System Sciences **68** (2004), no. 4, 841–860.

[Wil88] J.H. Wilkinson, *The algebraic eigenvalue problem (paperback ed.)*, Clarendon Press, Oxford, 1988.