Foundations and Trends® in
Theoretical Computer Science
Vol. 4, Nos. 3–4 (2008) 157–288

DOI: 10.1561/0400000025

now
the essence of knowledge

Spectral Algorithms

By Ravindran Kannan and Santosh Vempala

Contents

Foundations and Trends® in
Theoretical Computer Science
Vol. 4, Nos. 3–4 (2008) 157–288

DOI: 10.1561/0400000025

now
the essence of knowledge

Spectral Algorithms

Ravindran Kannan[1] and Santosh Vempala[2]

[1] *Microsoft Research, India, kannan@microsoft.com*
[2] *Georgia Institute of Technology, USA, vempala@cc.gatech.edu*

Abstract

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 as "continuous" problems. This monograph describes modern applications of spectral methods and novel algorithms for estimating spectral parameters. In the first part of the monograph, we present applications of spectral methods to problems from a variety of topics including combinatorial optimization, learning, and clustering. The second part of the monograph 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 monograph 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.

Part I

Applications

1

The Best-Fit Subspace

Many computational problems have explicit matrices as their input (e.g., adjacency matrices of graphs, experimental observations, etc.) while others refer to some matrix implicitly (e.g., document-term matrices, hyperlink structure, object–feature representations, network traffic, etc.). We refer to algorithms which use the spectrum, i.e., eigenvalues and vectors, singular values, and vectors, of the input data or matrices derived from the input as *Spectral Algorithms*. Such algorithms are the focus of this monograph. In the first part of this monograph, we describe applications of spectral methods in algorithms for problems from combinatorial optimization, learning, clustering, etc. In the second part, we study efficient randomized algorithms for computing basic spectral quantities such as low-rank approximations.

The Singular Value Decomposition (SVD) from linear algebra and its close relative, Principal Component Analysis (PCA), are central tools in the design of spectral algorithms. If the rows of a matrix are viewed as points in a high-dimensional space, with the columns being the coordinates, then SVD/PCA are typically used to reduce the dimensionality of these points, and solve the target problem in the lower-dimensional space. The computational advantages of such a

projection are apparent; in addition, these tools are often able to highlight hidden structure in the data. Section 1 provides an introduction to SVD via an application to a generalization of the least-squares fit problem. The next three chapters are motivated by one of the most popular applications of spectral methods, namely clustering. Section 2 tackles a classical problem from Statistics, learning a mixture of Gaussians from unlabeled samples; SVD leads to the current best guarantees. Section 3 studies spectral clustering for discrete random inputs, using classical results from random matrices, while Section 4 analyzes spectral clustering for arbitrary inputs to obtain approximation guarantees. In Section 5, we turn to optimization and see the application of tensors to solving maximum constraint satisfaction problems with a bounded number of literals in each constraint. This powerful application of low-rank tensor approximation substantially extends and generalizes a large body of work.

In the second part of this monograph, we begin with algorithms for matrix multiplication and low-rank matrix approximation. These algorithms (Section 6) are based on sampling rows and columns of the matrix from explicit, easy-to-compute probability distributions and lead to approximations additive error. In Section 7, the sampling methods are refined to obtain multiplicative error guarantees. Finally, in Section 8, we see an affine-invariant extension of standard PCA and a sampling-based algorithm for low-rank tensor approximation.

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-median problem. In one variant, the goal is to find a set of k

points that minimize the sum of the squared distances of the data points to their nearest facilities. A natural relaxation of this problem is to find the k-dimensional subspace for which the sum of the squared 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) and n corresponding eigenvectors.

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:

Proposition 1.1.

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

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,

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

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 $u = \frac{Av}{\sigma}$. it is easy to verify that u, v, and σ satisfy the singular value requirements.

The right singular vectors $\{v_i\}$ are thus exactly equal to the eigenvectors of A^TA. Since A^TA is a real, symmetric matrix, it has n orthonormal eigenvectors, which we can label $v_1, \ldots, v_n$. Expressing a unit vector v in terms of $\{v_i\}$ (i.e., $v = \sum_i \alpha_i v_i$ where $\sum_i \alpha_i^2 = 1$), we see that $\|Av\|^2 = \sum_i \sigma_i^2 \alpha_i^2$ which is maximized exactly when v corresponds to the top eigenvector of A^TA. If the top eigenvalue has multiplicity greater than 1, then v should belong to the space spanned by the top eigenvectors. □

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

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

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

Such a decomposition where,

1. The sequence of σ_is 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 $v_1, v_2, \ldots, v_n$ are all singular vectors. The assertion that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ follows from the definition of the v_is.

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 $v_1, v_2, \ldots, v_n$ 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 [48, 64, 67]. 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 1.5. Show that the power iteration applied k times to a symmetric matrix A finds a vector x^k such that

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

[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, $\mathsf{E}\left((x \cdot v)^2\right) = 1/n$].

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

1.3 The k-Variance 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". We call this the k-variance problem. The problem is NP-hard even for $k = 2$.

Note 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 1.6. 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-variance problem is thus the problem of partitioning a set of points into clusters so that the *sum of the variances of the clusters* is minimized.

We define a relaxation called the *Continuous Clustering Problem* (CCP), as the problem of finding the subspace V of $\mathbf{R}^n$ of dimension at most k which minimizes

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

The reader will recognize that this is given by the SVD. It is easy to see that the optimal value of the k-variance 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-variance problem using the relaxation to the best-fit subspace. The algorithm has two parts. First we project to the k-dimensional SVD subspace. Then we solve the problem in the smaller-dimensional space using a brute-force algorithm with the following guarantee.

Theorem 1.7. The k-variance problem can be solved in $O(m^{k^2d/2})$ 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 Equation (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- variance problem:

Algorithm: k-variance

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-variance 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-variance problem with input $\bar{\mathcal{A}}$.

Algorithm for the k-variance problem

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

It follows from Equation (1.2) that the optimal value $Z_{\mathcal{A}}$ of the k-variance 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-variance 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 Equation (1.3) we get

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

proving that we do indeed get a 2-approximation.

Theorem 1.8. Algorithm *k-variance* finds a factor-2 approximation for the k-variance problem 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-variance 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 general references on SVD/PCA, e.g., [12, 48].

The application of SVD to the k-variance problem is from [33] and its hardness is from [3]. 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 [58]; see also [49].

2

Mixture Models

This chapter is the first of three motivated by clustering problems. Here we study the setting 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 to 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 the next chapter, we move to discrete probability distributions, namely random graphs from some natural classes of distributions. In Section 4, we consider worst-case inputs and derive approximation guarantees for spectral clustering.

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 the component distributions (find their means, covariances, etc.).

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.

If the component distributions are far apart, so that points from one component distribution are closer to each other than to points from other components, then classification is straightforward. In the case of spherical Gaussians, making the means sufficiently far apart achieves this setting with high probability. On the other hand, if the component distributions have large overlap, then for a large fraction of the mixture, it is impossible to determine the origin of sample points. Thus, the classification problem is inherently tied to some assumption on the separability of the component distributions.

2.1 Probabilistic Separation

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

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

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. We can incorporate mixing weights in this condition, allowing for two components to overlap more if the mixing weight of one of them is small:

$$d_{TV}(f_1, f_2) = \int_{\boldsymbol{R}^n} |w_1 f_1(x) - w_2 f_2(x)| \, dx \geq 1 - \epsilon.$$

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.

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

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.

2.2 Geometric Separation

Here we assume some separation between the means of component distributions. 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.

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 2.1. 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 $\alpha > 1$,

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

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

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

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 $i \neq j$,

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

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. We do not describe this here.

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 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.

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

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.

2.3 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.
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 2.2. The best-fit one-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,

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

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 2.3. 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 2.4. The k-dimensional SVD subspace for a mixture of k Gaussians F contains the span of $\{\mu_1, \mu_2, \ldots, \mu_k\}$.

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!

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 2.4 extend to?
2. What about more general distributions?
3. What is the sample complexity?

2.4 Weakly Isotropic Distributions

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

Definition 2.1. Random variable $X \in \mathbb{R}^n$ has a *weakly isotropic* distribution with mean μ and variance σ^2 if

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

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

Exercise 2.5. Show that the uniform distribution in a cube is weakly isotropic.

Exercise 2.6. Show that a distribution is weakly isotropic if its covariance matrix is a multiple of the identity.

Exercise 2.7. The k-dimensional SVD subspace for a mixture F with component means $\mu_1, \ldots, \mu_k$ contains $\text{span}\{\mu_1, \ldots, \mu_k\}$ if each F_i is weakly isotropic.

The statement of Exercise 2.7 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 one-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.

2.5 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 two-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 cannot 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 2.8. 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 SVDsubspace 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 2.8 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 $\vec{e}_1, \ldots, \vec{e}_k$ be an orthonormal basis for W.

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

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 2.9. 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 $w_i = m_i/m$.

2.6 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 [55] holds for any logconcave density.

Lemma 2.10. 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,

$$\mathbf{Pr}(\|X-\mu\|^2 \geq tR) \leq e^{-t+1}.$$

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 2.11. 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.

2.7 An Affine-Invariant Algorithm

The algorithm described here is an application of isotropic PCA, an algorithm discussed in Section 8. 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 2.7.2 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 inter-mean 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.

2.7.1 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 intermean 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.

2.7.2 Analysis

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

Theorem 2.12. 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 2.12 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 [38, 44] 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

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

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 2.13. 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

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

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 $k > 2$ [15].

2.8 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 [22] in 1999, there has been much progress on efficient algorithms with rigorous guarantees [6, 23], with Arora and Kannan [6] addressing the case of general Gaussians using distance concentration methods. PCA was analyzed in this context by Vempala and Wang [65] giving nearly optimal guarantees for mixtures of spherical Gaussians (and weakly isotropic distributions). This was extended to general Gaussians and logconcave densities [51, 1] (Exercise 2.9 is based on [1]), 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 [15] presented an affine-invariant algorithm that only needs hyperplane separability for two Gaussians and a generalization of this condition for $k > 2$. A related line of work considers learning symmetric product distributions, where the coordinates are independent. Feldman et al. [39] 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 [17] have given a polynomial-time algorithm for clustering mixtures of product distributions (axis-aligned Gaussians) under mild separation conditions. A. Dasgupta et al. [21] and later Chaudhuri and Rao [18] gave algorithms for clustering mixtures of heavy-tailed distributions.

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 [14] 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 2.5.

3

Probabilistic Spectral Clustering

We revisit the problem of clustering under a model which assumes that the data is generated according to a probability distribution in $\boldsymbol{R}^n$. One line of work in this area pertains to mixture models where the components are assumed to have special distributions (e.g., Gaussians); in this situation, we saw in Section 2 that spectral methods are useful. Another line of work is based on models of random graphs. Typically, a random graph G on n vertices is assumed to be partitioned into k ($k << n$) unknown parts and an edge from a vertex in the r-th part to a vertex in the s-th part appears with probability p_{rs}, where these could be different for different r, s. The problem is to find the hidden partition and estimate the unknown p_{rs} values. Denoting by A the adjacency matrix of the 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).

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. Indeed, the literature often treats them as different subareas. In spite of this, as we will see in this chapter, spectral clustering can be used.

3.1 Full Independence and the Basic Algorithm

The basic tool which has been used to tackle the fat 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 3.1. 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 $c\nu\sqrt{n}$.[1]

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).

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

Thus one gets whp an upper bound on the spectral norm of $A - EA$:

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

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

Lemma 3.2. Suppose A, B are $m \times n$ matrices with $\text{rank}(B) = k$. If $\hat{A}$ is the best rank-k approximation to A, then

$$\|\hat{A} - B\|_F^2 \leq 5k\|A - B\|^2.$$

We use this 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 $< 1/(10k)$. The above implies that 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. We give a complete algorithm with a clean-up phase in Section . The algorithm is based only on linear algebraic assumptions rather than probabilistic ones.

We conclude this section with a proof of the lemma connecting the spectral norm and the Frobenius norm (from [1]).

Proof. (of Lemma 3.2): Let $u^{(1)}, u^{(2)}, \ldots u^{(k)}$ be the top k singular vectors of A. Extend this to an orthonormal basis $u^{(1)}, u^{(2)}, \ldots u^{(p)}$ of the vector space spanned by the rows of $\hat{A}$ and B. [Note that $p \leq 2k$.] Then, we have

$$\begin{aligned}
|\hat{A} - B\|_F^2 &= \sum_{t=1}^{k} |(\hat{A} - B)u^{(t)}|^2 + \sum_{t=k+1}^{p} |(\hat{A} - B)u^{(t)}|^2 \\
&= \sum_{t=1}^{k} |(A - B)u^{(t)}|^2 + \sum_{t=k+1}^{p} |Bu^{(t)}|^2 \\
&\leq k\|A - B\|_2^2 + \sum_{t=k+1}^{p} |Au^{(t)} + (B - A)u^{(t)}|^2 \\
&\leq k\|A - B\|_2^2 + 2\sum_{t=k+1}^{p} |Au^{(t)}|^2 + 2\sum_{t=k+1}^{p} |(B - A)u^{(t)}|^2 \\
&\leq k\|A - B\|_2^2 + 2k\sigma_{k+1}^2(A) + 2k\|A - B\|_2^2.
\end{aligned}$$

Now Lemma 3.2 follows from the claim : $\sigma_{k+1}(A) \leq \|A - B\|_2$. This is because, if not, letting now $v^{(1)}, v^{(2)}, \ldots v^{(k)}, v^{(k+1)}$ be the top $k+1$ singular vectors of A, we would have

$$|Bv^{(t)}| \geq |Av^{(t)}| - \|A - B\|_2 > 0,$$

contradicting the hypothesis that rank of B is k. □

3.2 Clustering Based on Deterministic Assumptions

We started earlier with a random generative model of data–A. We used Random Matrix theory to show a bound on $\|A - EA\|$. Then we argued that $\hat{A}$, the best rank-k approximation to A is in fact close to EA in spectral norm and used this to cluster "most" points correctly. However, the "clean-up" of the misclassified points presents a technical hurdle which is overcome often by extra assumptions and involved technical arguments. Here we make an attempt to present a simple algorithm which classifies all points correctly at once. We start by making certain assumptions on the model; these assumptions are purely geometric– we do not assume any probabilistic model. Under these assumptions, we prove that a simple algorithm correctly classifies all the points. A new feature of this proof is the use of the "Sin Θ" theorem from Numerical Analysis to argue that not only are the singular values of $\hat{A}$ and EA close, but the spaces spanned by these two matrices are close too. However, our result currently does not subsume earlier results under the probabilistic model. [See discussion below.]

We are given m points in $\mathbf{R}^n$ (as the rows of an $m \times n$ matrix A) and an integer k and we want to cluster (partition) the points into k clusters. As in generative models, we assume that there is an underlying (desirable) partition of $\{1, 2, \ldots m\}$ into $T_1, T_2, \ldots T_k$ which forms a "good" clustering and the objective is to find precisely this clustering (with not a single "misclassified" point). For $r = 1, 2, \ldots k$, define $\mu_r = \frac{1}{|T_r|} \sum_{i \in T_r} A_{(i)}$ as the center (mean) of the points in the cluster. Let C be the $m \times n$ matrix with $C_{(i)} = \mu_r$ for all $i \in T_r$. We will now state the assumptions under which we will prove that spectral clustering works. [We write assumptions of the form $a \in \Omega(b)$ below to mean that there is some constant $c > 0$ such that if the assumption $a \geq cb$

holds, then the assertions/algorithms work as claimed. Similarly for $a \in O(b)$.] We first assume

Assumption 0 :

$$\|A - C\| = \Delta \leq O(\sigma_k(C)/\log n).$$

[This is not a major assumption; see discussion below.] We note that $\|A - C\|^2$ can be viewed as the maximum total distance squared in any direction of the points from their respective centers. So Δ being small is the same as saying the displacements of $A_{(i)}$ from their respective centers are not "biased" toward any direction, but sort of spread out. [This is the intuition leading to Wigner-type bound on the largest singular value of a random matrix.]

Our main assumptions on the model are stated below.

Assumption 1 : Boundedness For all r and all $i \in T_r$,

$$|A_{(i)} - \mu_r| \leq M; \quad |\mu_r| \leq M.$$

Assumption 2 : Correct Center is closest. Let

$$\Delta_2 = \frac{M\Delta \log n}{\sigma_k(C)}.$$

Let F_1 be the orthogonal projection onto the space spanned by the rows of C. Then, for all $r \neq s$ and all $i \in T_r$,

$$|F_1(A_{(i)} - \mu_r)| \leq |F_1(A_{(i)} - \mu_s)| - \Omega(\Delta_2).$$

Assumption 3 : No Small Clusters

$$|T_r| \geq m_0 \in \Omega(m) \quad \forall r.$$

Note that Assumption 2 implies an **inter-center separation**

$$|\mu_r - \mu_s| = \Omega(\Delta_2).$$

Such an assumption is a regular feature of most results.

Now consider the random case when the A_{ij} are Bernoulli random variables with $EA_{ij} = C_{ij}$.(the Full-Independent case). For ease of comparison, assume $m \in \Theta(n)$ and that all (most) C_{ij} are $\Theta(p)$ for a positive real p. In this case, it is easy to see that we can take $M \in \tilde{\Theta}(\sqrt{np})$. Also

Random Matrix Theory implies that $\Delta \in \Theta(\sqrt{np})$. We also a need a lower bound on $\sigma_k(C)$ or in other words, we need C have rank k. We assume that $\sigma_k(C) = \Omega(np)$.

Thus $\Delta_2 = \tilde{O}(1)$. The best-known results for probabilistic models assume a separation of

$$|\mu_r - \mu_s| \geq \text{poly}(k)\sqrt{p}.$$

Thus our otherwise more general result does not match these.

We conjecture that the following clean result holds which would then subsume known previous results under various probabilistic models.

Conjecture We can exactly classify all points provided only the following assumption holds:

$$\forall r \neq s, \quad \forall i \in T_r,$$
$$|F_1(A_{(i)} - \mu_r)| \leq |F_1(A_{(i)} - \mu_s)| - \Omega\left(\text{poly}(k)\|A - C\|/\sqrt{n}\right).$$

3.2.1 The Algorithm

We use an approximation algorithm to solve the k-means problem on the points–$\hat{A}_{(i)}, i = 1, 2, \ldots m$ to within a factor of say c_2. A simple algorithm has been shown to achieve $c_2 \in O(\log n)$ [9], but $c_2 \in O(1)$ can be achieved by more complex algorithms [16].

Theorem 3.3. Under Assumptions (0)–(3), the algorithm finds the correct clustering, i.e., all i for which $A_{(i)}$ is closest to a particular row of C are put in the same cluster.

Suppose the centers produced by the approximation algorithm are $v_1, v_2, \ldots v_r$. Let $c_1 = 6\sqrt{c_2 + 2}$.

Note that the optimal k-means solution has optimal value OPT at most $\sum_i |\hat{A}_{(i)} - C_{(i)}|^2 = \|\hat{A} - C\|_F^2$.

Claim 3.4. In a c_2-approximate solution, we must have that for each $r, 1 \leq r \leq k$, there is a center v_{i_r} (in the solution) such that $|v_{i_r} - \mu_r| \leq \frac{c_1\sqrt{k}}{\sqrt{m_0}}\|A - C\|$.

Proof. Let $\frac{c_1\sqrt{k}}{\sqrt{m_0}}\|A-C\| = \beta$. Suppose for some r, there is no center in the solution within distance β of μ_r. Then we have using triangle inequality and the fact that $(a-b)^2 \geq \frac{1}{2}a^2 - b^2$ for any reals a, b that the sum of distances squared of $\hat{A}_{(i)}, i \in T_r$ to their nearest center in the solution is at least

$$\sum_{i\in T_r}(\beta - |\hat{A}_{(i)} - \mu_r|)^2 \geq (|T_r|/2)\beta^2 - \|\hat{A}-C\|_F^2 > c_2 \text{ OPT}$$

producing a contradiction. □

Now $\sigma_k(C) \leq \frac{1}{\sqrt{k}}\|C\|_F \leq \frac{\sqrt{m}}{\sqrt{k}}M$; thus, $\frac{\sqrt{k}}{\sqrt{m}}\Delta \in O(\Delta_2)$. Thus, for a suitable choice of c_1, c_2, there must be k different v_r; for notational convenience, we assume from now on that

$$|v_r - \mu_r| \in O(\Delta_2). \tag{3.1}$$

Let

$$S_r = \{i : |\hat{A}_{(i)} - v_r| \leq |\hat{A}_{(i)} - v_s| \forall s\}.$$

Now, we will argue using the assumption that S_r is exactly equal to T_r for all r.

To this end let F_2 denote (orthogonal) projection onto the space spanned by the top k right singular vectors of A and recall that F_1 denotes the orthogonal projection onto the space spanned by the rows of C. We argue that $F_1 \approx F_2$ using Davis–Kahan Sinθ theorem. The theorem applies to Hermitian matrices. Of course A, C are in general rectangular. So first let $|A|$ denote $\sqrt{A^TA}$ and similarly $|C|$ denote $\sqrt{C^TC}$ (standard notation). It is known ([11], Equation (5.10)) that there is a fixed constant with

$$|| \, |A| - |C| \, || \leq c_3 \log n \|A-C\|.$$

Clearly $\sigma_k(A) \geq \sigma_k(C) - \|A-C\| \geq \frac{1}{2}\sigma_k(C)$. $F_1^{\perp}$ can be viewed as the projection onto the eigenvectors of $|C|$ with eigenvalues less than or equal to 0. Now we know ([12] Exercise VII.1.11 and the sine θ theorem: Theorem VII.3.1)

$$\|F_1^{\perp}F_2\| = \|F_2 - F_1\| \leq \frac{c_4 \log n \Delta}{\sigma_k(C)} \in O(\Delta_2/M). \tag{3.2}$$

Now we use this as follows: for any $r \neq s$ and $i \in T_r$,

$$\begin{aligned}
|F_2(A_{(i)} - v_r)| &\leq |F_2(A_{(i)} - \mu_r)| + |F_2(\mu_r - v_r)| \\
&\leq |F_1(A_{(i)} - \mu_r)| + O(\Delta_2) + |v_r - \mu_r| \text{ Assumption 1} \\
&\quad \text{and Equation (3.2)} \\
&\leq |F_1(A_{(i)} - \mu_s)| - \Omega(\Delta_2) \text{ Assumption 2} \\
&\leq |F_2(A_{(i)} - \mu_s)| - \Omega(\Delta_2) \text{ using Equation (3.2)} \\
&\quad \text{provided } |A_{(i)} - \mu_s| \in O(M) \\
&\leq |F_2(A_{(i)} - v_s)| - \Omega(\Delta_2) \text{ using Equation (3.1)}
\end{aligned}$$

Now if $|A_{(i)} - \mu_s| \geq 10M$, then we argue differently. First we have

$$\begin{aligned}
|F_1(A_{(i)} - \mu_s)|^2 &= |A_{(i)} - \mu_s|^2 - |A_{(i)} - F_1(A_{(i)})|^2 \\
&\geq |A_{(i)} - \mu_s|^2 - |A_{(i)} - \mu_r|^2.
\end{aligned}$$

Thus, $|F_1(A_{(i)} - \mu_s)| \geq 0.9|A_{(i)} - \mu_s|$. So we have (recalling Assumption (0))

$$\begin{aligned}
|F_2(A_{(i)} - \mu_s)| &\geq |F_1(A_{(i)} - \mu_s)| - |A_{(i)} - \mu_s|\frac{\Delta_2}{M} \\
&\geq 0.8|A_{(i)} - \mu_s| \\
&\geq |A_{(i)} - \mu_r|.
\end{aligned}$$

3.3 Proof of the Spectral Norm Bound

Here we prove Wigner's theorem (Theorem 3.1) 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 3.5. Any vector $u \in \boldsymbol{R}^n$ with $\|u\| = 1$ can be written as

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

where

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

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,

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

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

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

and

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

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

$$\begin{aligned}
\|\frac{u_k}{r^k}\| \leq \|u - \sum_{i=0}^{k-1}\left(\frac{1}{r}\right)^i u_i\| + diag(\mathcal{L}/r^k) &\leq \frac{1}{r^k} + \frac{1}{r^{k+1}} \\
\implies \|u_k\| &\leq 1 + \frac{1}{r} \\
\|u - \sum_{i=0}^{k}\left(\frac{1}{r}\right)^i u_i\| &\leq \frac{1}{r^{k+1}} \longrightarrow 0.
\end{aligned}$$

That completes the proof. □

Now using Lemma 3.5, we will show that it suffices to consider only the lattice vectors in $\mathcal{L} \cap \mathbb{B}(\bar{0}, 1 + 1/r)$ instead of all unit vectors in order to bound $\lambda(A)$. Indeed, this bound holds for the spectral norm of a tensor.

Proposition 3.6. For any matrix A,

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

Proof. From Lemma 3.5, we can write any u with $\|u\| = 1$ as

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

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

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

which proves the proposition. □

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

Lemma 3.7. The number of lattice points in $\mathcal{L} \cap \mathbb{B}(\bar{0}, 1 + 1/r)$ 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,

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

□

The following Chernoff bound will be used.

Exercise 3.8. 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 $u, v \in \mathcal{L} \cap \mathbb{B}(\bar{0}, 1 + 1/r)$. 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

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

So using $t = 4\sqrt{n}\sigma$ in the Chernoff bound Equation (3.8),

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

According to Lemma 3.7, there are at most $(2r)^{2n}$ ways of picking $u, v \in \mathcal{L} \cap \mathbb{B}(\bar{0}, 1 + 1/r)$. so we can use union bound to get

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

for $r = 2$. And finally using Proposition 3.6 and the facts that for our choice of r, $\sigma \leq 9/4$ and $(r/r-1)^2 \leq 4$, we have

$$\mathbf{Pr}\left(\lambda(A) \geq 36\sqrt{n}\right) \leq e^{-5n}.$$

This completes the proof. □

The above bound can be extended to r-dimensional tensors.

Exercise 3.9. 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 $A\left(u^{(1)}, u^{(2)}, \ldots, u^{(r)}\right) = \sum_{i_1,i_2,\ldots,i_r} A_{(i_1,i_2,\ldots,i_r)} \quad u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(r)}_{i_r}$. Suppose each entry of A is 1 or -1 with equal probability. Show that whp,

$$\lambda(A) = O(\sqrt{n} r \log r). \tag{3.3}$$

3.4 Discussion

The bounds on eigenvalues of symmetric random matrices, formulated by Wigner, were proved by Füredi and Komlos [45] and tightened by Vu [66]. 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 [13] and Alon et al. [5]. Subsequently, McSherry gave a simpler algorithm for finding planted partitions [57]. Spectral projection was also used in random models of information retrieval by Papadimitriou et al. [59] and extended by Azar et al. [10].

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. Dasgupta et al. [20] give bounds for spectral norms of such matrices based on the functional analysis work of Rudelson [60] and Lust-Piquard [56]. It is an open problem to give a simple, optimal clean-up algorithm for probabilistic spectral clustering.

4

Recursive Spectral Clustering

In this chapter, we study a spectral algorithm for 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. Given a graph $G = (V, E)$, with non-negative 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

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

4.1 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 adjacency matrix so each row sum is 1.
2. Find the second largest eigenvector of this matrix.
3. Order the vertices according to 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 4.1. Suppose B is an $N \times N$ 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

$$\min_{S \subseteq \{1,2,\ldots N\}} \frac{\sum_{i \in S, j \notin S} \pi_i b_{ij}}{\min\left(\sum_{i \in S} \pi_i, \sum_{j \notin S} \pi_j\right)}$$

$$\geq 1 - \lambda_2 \geq \frac{1}{2} \left(\min_{l, 1 \leq l \leq N} \frac{\sum_{1 \leq u \leq l; l+1 \leq v \leq N} \pi_{i_u} b_{i_u i_v}}{\min\left(\sum_{1 \leq u \leq l} \pi_{i_u}, \sum_{l+1 \leq v \leq N} \pi_{i_v}\right)} \right)^2$$

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

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

Proof (of Theorem 4.1). We first evaluate the second eigenvalue. Toward this end, let $D^2 = \text{diag}(\pi)$. Then, from the time-reversibility property of B, we have $D^2B = B^TD^2$. Hence $Q = DBD^{-1}$ is symmetric. The eigenvalues of B and Q are the same, with their largest eigenvalue equal to 1. In addition, $\pi^TD^{-1}Q = \pi^TD^{-1}$ and therefore π^TD^{-1} is the left eigenvector of Q corresponding to the eigenvalue 1. So we have,

$$\lambda_2 = \max_{\pi^TD^{-1}x=0} \frac{x^TDBD^{-1}x}{x^Tx}$$

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

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

The numerator can be rewritten as

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

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

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

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

$$w_i = \begin{cases} \sqrt{\frac{1}{\sum_u a(u)} \frac{\pi(\bar{S})}{\pi(S)}} & \text{if } i \in S \\ -\sqrt{\frac{1}{\sum_u a(u)} \frac{\pi(S)}{\pi(S)}} & \text{if } i \in \bar{S} \end{cases}$$

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

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

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

$$\begin{aligned}
&\left(\sum_{i<j}\pi_i b_{ij}(z_i - z_j)^2\right)\left(\sum_{i<j}\pi_i b_{ij}(|z_i| + |z_j|)^2\right) \\
&\quad\geq \left(\sum_{i<j}\pi_i b_{ij}|z_i - z_j|(|z_i| + |z_j|)\right)^2 \\
&\quad\geq \left(\sum_{i<j}\pi_i b_{ij}\sum_{k=i}^{j-1}|z_{k+1}^2 - z_k^2|\right)^2 \qquad (4.1)
\end{aligned}$$

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

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

Since $z_r = 0$, we obtain

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

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

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

4.2 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 [52].

Definition 4.1. 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.

4.3 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 [46]. 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 0 and 1. The following theorem provides a guarantee for the approximate-cluster algorithm using $\mathcal{A}$ as a subroutine.

Theorem 4.2. 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 $\alpha^* = \frac{\alpha}{6 \log n/\epsilon}$. 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 $w_{\text{I}}(S_j,T_j)$ the sum of the weights of the intra-cluster edges of the cut (S_j,T_j), i.e., $w_{\text{I}}(S_j,T_j) = \sum_{i=1}^{l} w(S_j \cap C_i, T_j \cap C_i)$. We then set

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

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

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

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,

$$w(S_j,T_j) \leq K\big(2w(P(S'_j))\big)^{\nu} a(S_j)^{1-\nu} \tag{4.2}$$

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:

$$\begin{aligned}
\sum_{j \in H} w(S_j,T_j) &\leq \sum_{\text{all } j} K\big(2w(P(S'_j))\big)^{\nu} a(S_j)^{1-\nu} \\
&\leq K\left(2\sum_{\text{all } j} w(P(S'_j))\right)^{\nu} \left(\sum_j a(S_j)\right)^{1-\nu} \\
&\leq K\left(2\epsilon \log \frac{n}{\epsilon}\, a(V)\right)^{\nu} \left(2\log \frac{n}{\epsilon}\, a(V)\right)^{1-\nu} \\
&\leq 2K\epsilon^{\nu} \log \frac{n}{\epsilon}\, a(V)
\end{aligned} \tag{4.3}$$

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 belongs 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 endpoints in different sets of the partition. So, given that C_i has conductance α, we obtain

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

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

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

Thus,

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

Therefore, from the definition of H, we have

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

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 Equation (4.3) then gives

$$\sum_{j \notin H} w_{\text{I}}(S_j, T_j) \leq 2 \sum_{j \in H} w_{\text{I}}(S_j, T_j) \leq 4K\epsilon^{\nu} \log \frac{n}{\epsilon} a(V) \tag{4.4}$$

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

$$\sum_{j \notin H} (w(S_j, T_j) - w_{\text{I}}(S_j, T_j)) \leq \frac{\epsilon}{2} a(V) \tag{4.5}$$

We then sum up Equations (4.3)–(4.5). 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 4.2.

The Leighton–Rao algorithm for approximating the conductance finds a cut of conductance at most $2 \log n$ times the minimum [54]. In our terminology, it is an approximation algorithm with $K = 2 \log n$ and $\nu = 1$. Applying Theorem 4.2 leads to the following guarantee.

Corollary 4.3. 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 [8] of $O(\sqrt{\log n})$.

Corollary 4.4. 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.

4.4 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 minimizes $\phi(\{u_1, u_2, \ldots u_j\}, C_t)$ 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 4.1 with Theorem 4.2 to get a worst-case guarantee for Algorithm Recursive-Spectral. In the terminology of Theorem 4.2, Theorem 4.1 says that the spectral heuristic for minimum conductance is an approximation algorithm with $K = \sqrt{2}$ and $\nu = 1/2$.

Corollary 4.5. If the input has an (α, ϵ)-clustering, then, using the spectral heuristic, the approximate-cluster algorithm finds an

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

4.5 Discussion

This chapter is based on Kannan et al. [52] and earlier work by Sinclair and Jerrum [62]. Theorem 4.1 was essentially proved by Sinclair and Jerrum (in their proof of Lemma 3.3 in [62], although not mentioned in

the statement of the lemma). Cheng et al. [19] 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 [63] 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 [53].

5

Optimization via Low-Rank Approximation

In this chapter, we study Boolean constraint satisfaction problems (CSPs) with r variables per constraint. The general problem is weighted MAX-rCSP: given an rCSP with a weight for each constraint, find a Boolean assignment that maximizes the total weight of satisfied constraints. This captures numerous interesting special cases, including problems on graphs such as max-cut. We study an approach based on low-rank tensor approximation, i.e., approximating a tensor (multi-dimensional array) by the sum of a small number of rank-1 tensors. An algorithm for efficiently approximating a tensor by a small number of rank-1 tensors is given in Section 8. Here we apply it to the max-rCSP problem and obtain a polynomial-time approximation scheme under a fairly general condition (capturing all known cases).

A MAX-rCSP problem can be formulated as a problem of maximizing a homogenous degree r polynomial in the variables $x_1, x_2, \ldots x_n$, $(1-x_1), (1-x_2), \ldots (1-x_n)$ (see, e.g., [4].) Let

$$\mathbf{S} = \{y = (x_1, \ldots x_n, (1-x_1), \ldots (1-x_n)) : x_i \in \{0,1\}\}$$

be the solution set. Then the problem is

$$\text{MAX}_{y\in\mathbf{S}} \sum_{i_1,i_2,\ldots i_r=1}^{2n} A_{i_1,i_2,\ldots i_r} y_{i_1} y_{i_2} \ldots y_{i_r},$$

where A is a given non-negative symmetric r-dimensional array, i.e.,

$$A_{i_1,i_2,\ldots i_r} = A_{i_{\sigma(1)},i_{\sigma(2)},\ldots i_{\sigma(r)}}$$

for any permutation σ. The entries of the r-dimensional array A can be viewed as the weights of an r-uniform hypergraph on $2n$ vertices. Throughout, we assume that r is fixed.

Our main tool to solve this problem is a generalization of low-rank matrix approximation. A rank-1 tensor is the *outer product* of r vectors $x^{(1)},\ldots x^{(r-1)},x^{(r)}$, given by the r-dimensional array whose $(i_1,\ldots i_r)$'th entry is $x^{(1)}_{i_1}x^{(2)}_{i_2},\ldots x^{(r)}_{i_r}$; it is denoted $x^{(1)}\otimes x^{(2)}\otimes\ldots x^{(r)}$.

In Section 8, it is shown that

1. For any r-dimensional array A, there exists a good approximation by the sum of a small number of rank-1 tensors (Lemma 8.1).
2. We can algorithmically find such an approximation (Theorem 8.2).

In the case of matrices, traditional Linear Algebra algorithms find good approximations. Indeed, we can find the *best* approximations under both the Frobenius and L_2 norms using the Singular Value Decomposition. Unfortunately, there is no such theory for r-dimensional arrays when $r \geq 2$. Nevertheless, the sampling-based algorithm from Section 8 will serve our purpose.

We conclude this section by defining two norms of interest for tensors, the Frobenius norm and the 2-norm, generalizing the corresponding norms for matrices.

$$\|A\|_F = \left(\sum A^2_{i_1,i_2,\ldots i_r}\right)^{\frac{1}{2}}$$

$$\|A\|_2 = \max_{x^{(1)},x^{(2)},\ldots x^{(r)}} \frac{A(x^{(1)},x^{(2)},\ldots x^{(r-1)},x^{(r)})}{|x^{(1)}||x^{(2)}|\ldots}.$$

5.1 A Density Condition

We begin with a density condition on tensors. We will see later that if a MAX-rCSP viewed as a weighted r-uniform hypergraph satisfies this condition, then there is a PTAS for the problem. This condition provides a unified framework for a large class of weighted MAX-rCSPs.

Define the node weights $D_1, \ldots, D_{2n}$ of A and their average as

$$D_i = \sum_{i_2, i_3, \ldots i_r \in V} A_{i, i_2, \ldots i_r} \qquad \bar{D} = \frac{1}{2n} \sum_{i=1}^{n} D_i.$$

Note that when $r = 2$ and A is the adjacency matrix of a graph, the D_i are the degrees of the vertices and $\bar{D}$ is the average degree.

Definition 5.1. The *core-strength* of a weighted r-uniform hypergraph given by an r-dimensional tensor A is

$$\left(\sum_{i=1}^{2n} D_i \right)^{r-2} \sum_{i_1, i_2, \ldots, i_r} \frac{A^2_{i_1, \ldots, i_r}}{\prod_{j=1}^{r} (D_{i_j} + \bar{D})}$$

We say that a class of weighted hypergraphs (MAX-rCSPs) is *core-dense* if the core-strength is $O(1)$ (i.e., independent of A, n).

To motivate the definition, first suppose the class consists of unweighted hypergraphs. Then if a hypergraph in the class has E as the edge set with $|E| = m$ edges, the condition says that (for any constant r),

$$m^{r-2} \sum_{(i_1, \ldots, i_r) \in E} \frac{1}{\prod_{j=1}^{r} (D_{i_j} + \bar{D})} = O(1). \tag{5.1}$$

Note that here the D_is are the degrees of the hypergraph vertices in the usual sense of the number of edges incident to the vertex. It is easy to see this condition is satisfied for dense hypergraphs, i.e., for r-uniform hypergraphs with $\Omega(n^r)$ edges, because in this case, $\bar{D} \in \Omega(n^{r-1})$. The dense case was the first major milestone of progress on this problem.

The condition can be specialized to the case $r = 2$, where it says that

$$\sum_{i,j} \frac{A^2_{ij}}{(D_i + \bar{D})(D_j + \bar{D})} = O(1). \tag{5.2}$$

We will show that all metrics satisfy this condition. Also, so do *quasi-metrics*. These are weights that satisfy the triangle inequality up to a constant factor (e.g., powers of a metric). So a special case of the main theorem is a PTAS for metrics and quasi-metrics. The main result of this chapter is the following.

Theorem 5.1. There is a PTAS for any core-dense weighted MAX-rCSP.

The algorithm and proof are given in Section 5.3. We will also show (in Section 5.4) that a generalization of the notion of metric for higher r also satisfies our core-dense condition.

Theorem 5.2. Suppose for a MAX-rCSP, the tensor A satisfies the following local density condition:

$$\forall i_1, \ldots, i_r, \quad A_{i_1,\ldots,i_r} \leq \frac{c}{n^{r-1}} \sum_{j=1}^{r} D_{i_j}$$

where c is a constant. Then there is a PTAS for the MAX-rCSP defined by A.

The condition in the theorem says that no entry of A is "wild" in that it is at most a constant times the average entry in the r "planes" passing through the entry. The reason for calling such tensors "metric tensors" will become clear when we see in Section 5.4 that for $r = 2$, metrics do indeed satisfy this condition. When the matrix A is the adjacency matrix of a graph, then the condition says that for any edge, one of its end points must have degree $\Omega(n)$. This is like the "everywhere dense" condition in [7]. Theorem 5.2 has the following corollary for "quasi-metrics", where the triangle inequality is only satisfied within constant factors - $A_{ik} \leq c(A_{ij} + A_{jk})$.

Corollary 5.3. There exists a PTAS for metric and quasimetric instances of MAX-CSP.

5.2 The Matrix Case: MAX-2CSP

In this section, we prove Theorem 5.1 in the case $r = 2$. This case already contains the idea of scaling which we will use for the case of higher r. However, this case does not need new algorithms for finding low-rank approximations as they are already available from classical linear algebra.

Recall that we want to find

$$\text{MAX}_{y \in \mathbf{S}} A_{ij} y_i y_j = y^T A y,$$

where

$$\mathbf{S} = \{y = (x_1, x_2, \ldots x_n, (1 - x_1), (1 - x_2), \ldots (1 - x_n)), x_i \in \{0, 1\}\}$$

is the solution set. We will describe in this section an algorithm to solve this problem to within additive error $O(\epsilon n \bar{D})$, under the assumption that that the core-strength of A is at most a constant c. The algorithm will run in time polynomial in n for each fixed $\epsilon > 0$. Note that

$$\text{MAX}_{y \in \mathbf{S}} y^T A y \geq \mathsf{E}\,(y^T A y) = \frac{1}{2} n \bar{D},$$

where E denotes expectation over uniform random choice of $x \in \{0, 1\}^n$. Thus, this will prove Theorem 5.1 for this case (of $r = 2$).

In the algorithm below for MAX-2CSP, we assume the input is a matrix A whose entries denote the weights of the terms in the CSP instance.

Algorithm: Approximate MAX-2CSP

1. Scale the input matrix A as follows:
$$B = D^{-1} A D^{-1}$$
where D is the diagonal matrix with $D_{ii} = \sqrt{D_i + \bar{D}}$.
2. Find a low-rank approximation $\hat{B}$ to B such that
$$\|B - \hat{B}\|_2 \leq \frac{\epsilon}{2} \|B\|_F$$
and rank of $\hat{B}$ is $O(1/\epsilon^2)$.
3. Set $\hat{A} = D \hat{B} D$.
4. Solve $\max_{y \in S} y^T \hat{A} y$ approximately.

The last step above will be expanded presently. We note here that it is a low-dimensional problem since $\hat{A}$ is a low-rank matrix.

In the first step, the algorithm scales the matrix A. A related scaling,

$$B_{ij} = \frac{A_{ij}}{\sqrt{D_i}\sqrt{D_j}}$$

is natural and has been used in other contexts (for example when A is the transition matrix of a Markov chain). This scaling unfortunately scales up "small degree" nodes too much for our purpose and so we use a modified scaling. We will see that while the addition of $\bar{D}$ does not increase the error in the approximation algorithms, it helps by modulating the scaling up of low degree nodes. From the definition of core-strength, we get the next claim.

Claim 5.4. $\|B\|_F^2$ is the core-strength of the matrix A.

The second step is performed using the SVD of the matrix B in polynomial-time. In fact, as shown in [43], such a matrix $\hat{B}$ can be computed in linear in n time with error at most twice as large.

After the third step, the rank of $\hat{A}$ equals the rank of $\hat{B}$. In the last step, we solve the following problem approximately to within additive error $O(\epsilon n\bar{D})$:

$$\max_{y \in \mathbf{S}} y^T \hat{A} y \tag{5.3}$$

We will see how to do this approximate optimization presently. First, we analyze the error caused by replacing A by $\hat{A}$.

$$\begin{aligned}
\text{MAX}_{y\in\mathbf{S}}|y^T(A-\hat{A})y| &= \text{MAX}_{y\in\mathbf{S}}|y^T D(B-\hat{B})Dy| \\
&\leq \text{MAX}_{y\in\mathbf{S}}|Dy|^2\|B-\hat{B}\|_2 \\
&\leq \epsilon\sum_i (D_i+\bar{D})\|B\|_F \\
&\leq 4\epsilon n\bar{D}(\text{core-strength of } A)^{1/2},
\end{aligned}$$

the last because of Claim 5.4 and the fact that $\sum_i D_i = 2n\bar{D}$.

Now for solving the non-linear optimization Problem (5.3), we proceed as follows: suppose the SVD of $\hat{B}$ expressed $\hat{B}$ as $U\Sigma V$, where the

U is an $2n \times l$ matrix with orthonormal columns, Σ is a $l \times l$ diagonal matrix with the singular values of $\hat{B}$ and V is a $l \times 2n$ matrix with orthonormal rows. We write

$$y^T \hat{A} y = (y^T DU)\Sigma(VDy) = u^T \Sigma v$$
$$\text{where, } u^T = y^T DU \quad \text{and} \quad v = VDy$$

are two l- vectors. This implies that there are really only $2l$ "variables"– u_i, v_i in the problem (and not the n variables–$y_1, y_2, \ldots y_n$). This is the idea we will exploit. Note that for $y \in \mathbf{S}$, we have (since U, V have orthonormal columns, rows, respectively)

$$|u|^2 \leq |y^T D|^2 \leq \sum_i (D_i + \bar{D}) \leq 4n\bar{D}.$$

Similarly, $|v|^2 \leq 4n\bar{D}$. So letting

$$\alpha = \sqrt{n\bar{D}},$$

we see that the vectors u, v live in the rectangle

$$R = \{(u, v) : -2\alpha \leq u_i, v_j \leq +2\alpha\}.$$

Also, the gradient of the function $u^T \Sigma v$ with respect to u is Σv and with respect to v is $u^T \Sigma$; in either case, the length of the gradient vector is at most $2\alpha\sigma_1(\hat{B}) \leq 2\alpha\sqrt{c}$. We now divide up R into small cubes; each small cube will have side

$$\eta = \frac{\epsilon\alpha}{20\sqrt{l}},$$

and so there will be $\epsilon^{-O(l)}$ small cubes. The function $u^T \Sigma v$ does not vary by more than $\epsilon n\bar{D}\sqrt{c}/10$ over any small cube. Thus we can solve Equation (5.3) by just enumerating all the small cubes in R and for each determining whether it is feasible (i.e., whether there exists a 0–1 vector x such that for some (u, v) in this small cube, we have $u^T = y^T Du, v = VDy$, for $y = (x, \underline{1} - x)$).

For each small cube C in R, this is easily formulated as an integer program in the n 0,1 variables $x_1, x_2, \ldots x_n$ with $4l$ constraints (arising from the upper and lower bounds on the coordinates of u, v which ensure that (u, v) is in the small cube.)

For a technical reason, we have to define a D_i to be "exceptional" if $D_i \geq \epsilon^6 n\bar{D}/10^6$; also call an i exceptional if either D_i or D_{i+n} is exceptional. Clearly, the number of exceptional D_i is at most $2 \times 10^6/\epsilon^6$ and we can easily identify them. We enumerate all possible sets of $2^{O(1/\epsilon^6)}$ 0,1 values of the exceptional x_i and for each of these set of values, we have an Integer Program again, but now only on the non-exceptional variables.

We consider the Linear Programming (LP) relaxation of each of these Integer Programs obtained by relaxing $x_i \in \{0,1\}$ to $0 \leq x_i \leq 1$. If one of these LPs has a feasible solution, then, it has a basic feasible solution with at most $4l$ fractional variables, Rounding all these fractional variables to 0 changes Dy by a vector of length at most

$$\sqrt{4l\epsilon^6 n\bar{D}/10^6} \leq \eta.$$

Thus, the rounded integer vector y gives us a (u, v) in the small cube C enlarged (about its center) by a factor of 2 (which we call $2C$). Conversely, if none of these LPs has a feasible solution, then clearly neither do the corresponding Integer Programs and so the small cube C is infeasible. Thus, for each small cube C, we find (i) either C is infeasible or (ii) $2C$ is feasible. Note that $u^T \Sigma v$ varies by at most $\epsilon n\bar{D}/5$ over $2C$. So, it is clear that returning the maximum value of $u^T \Sigma v$ over all centers of small cubes for which (ii) holds suffices.

We could have carried this out with any "scaling". The current choice turns out to be useful for the two important special cases here. Note that we are able to add the $\bar{D}$ almost "for free" since we have $\sum_i D_i + \bar{D} \leq 2\sum D_i$.

5.3 MAX-rCSPs

In this section, we consider the general case of weighted MAX-rCSPs and prove Theorem 5.1. The algorithm is a direct generalization of the two-dimensional case.

For any k vectors $x^{(1)}, x^{(2)}, \ldots x^{(k)}$, the $r-k$-dimensional tensor

$$A(x^{(1)}, x^{(2)}, \ldots x^{(k)}, \cdot, \cdot) = \sum_{i_1, i_2, \ldots i_{r-1}} A_{i_1, i_2, \ldots i_{r-1}, i} x^{(1)}_{i_1} x^{(2)}_{i_2}, \ldots x^{(r-1)}_{i_{r-1}}.$$

We wish to solve the problem

$$\max_{y \in \mathbf{S}} A(y, y, \ldots, y).$$

Algorithm: Approximate MAX-rCSP

1. Scale the input tensor A as follows:

$$B_{i_1,\ldots,i_r} = \frac{A_{i_1,\ldots,i_r}}{\prod_{j=1}^{r} \alpha_{i_j}},$$

   where $\alpha = (\alpha_1, \ldots, \alpha_n) \in \boldsymbol{R}^n$ is defined by $\alpha_j = \sqrt{\bar{D} + D_j}$.

2. Find a tensor $\hat{B}$ of rank at most k satisfying

$$\|B - \hat{B}\|_2 \leq \frac{\epsilon}{2} \|B\|_F.$$

3. Let $z_j = y_j \alpha_j$, for $y \in S$, so that

$$A(y, \ldots, y) = B(z, \ldots, z).$$

4. Solve

$$\max_{z : y_j \in \mathbf{S}_1} \hat{B}(z, z, \ldots, z)$$

   to within additive error $\epsilon |\alpha|^r \|B\|_F / 2$.

The error of approximating B by $\hat{B}$ is bounded by

$$\begin{aligned}
\max_{z \in \mathbf{S}_1} |(B - \hat{B})(z, \ldots, z)| &\leq \max_{z : |z| \leq |\alpha|\}} |(B - \hat{B})(z, \ldots, z)| \\
&\leq |\alpha|^r \|B - \hat{B}\|_2 \\
&\leq \epsilon |\alpha|^r \|B\|_F \\
&\leq \epsilon \left(\sum_{i=1}^{n} (\bar{D} + D_i) \right)^{r/2} \left(\sum_{i_1,\ldots,i_r} \frac{A_{i_1,\ldots,i_r}^2}{\prod_{j=1}^{r} D_{i_j}} \right)^{1/2} \\
&\leq \epsilon 2^{r/2} c \left(\sum_{i=1}^{n} D_i \right)
\end{aligned}$$

where c is the bound on the core-strength, noting that $\sum_i(\bar{D} + D_i) = 2\sum_i D_i$.

5.3.1 Optimizing Constant-Rank Tensors

From the above it suffices to deal with a tensor of constant rank. Let A be a tensor of dimension r and rank ℓ, say:

$$A = \sum_{1 \le j \le \ell} A^{(j)}$$

with

$$A^{(j)} = a_j x^{(j,1)} \otimes x^{(j,2)} \cdots \otimes x^{(j,r)}$$

where the $x^{(j,i)} \in \mathbf{R}^{2n}$ are length one vectors and moreover we have that $\|A^{(j))}\|_F \le \|A\|_F$ and $\ell = O(\epsilon^{-2})$. We want to maximize approximately $B(y, y, \cdots y)$, over the set of vectors y satisfying for each $i \le n$ either $(y_i, y_{n+i}) = (0, \alpha_{n+i})$ or $(y_i, y_{n+i}) = (\alpha_i, 0)$ where α is a given $2n$-dimensional positive vector. Let us define the tensor B by

$$B_{i_1,i_2,\ldots,i_r} = \alpha_{i_1}\alpha_{i_2}, \ldots, \alpha_{i_r} A_{i_1,i_2,\ldots,i_r} \ \forall\ i_1, i_2, \ldots, i_r \in V.$$

Then, with $y_j = \alpha_j x_j$, we have that

$$B(x, x, \ldots, x) = A(y, y, \ldots, y).$$

Thus, we can as well maximize approximately B now for y in $\mathbf{S}$. We have

$$B(y, y, \cdots y) = \sum_{j=1}^{\ell} a_j \left(\prod_{k=1}^{r} (z^{(j,k)} \cdot y \right) \tag{5.4}$$

with

$$z^{(j,r)} = \alpha^T x^{(j,r)},\ 1 \le j \le \ell,\ 1 \le k \le r.$$

Similarly as in the two-dimensional case, $B(y, y, \cdots y)$ depends really only on the ℓr variables $u_{j,i}$, say, where $u_{j,i} = z^{(j,i)} \cdot y,\ j = 1, 2, \ldots, \ell,\ i = 1, 2, \ldots, r$, and the values of each of these products are confined to the interval $[-2|\alpha|, +2|\alpha|]$. Then, exactly similarly as in the

two-dimensional case, we can get in polynomial-time approximate values for the $u_{j,i}$ within $\epsilon|\alpha|$ from the optimal ones. Inserting then these values in Equation (5.4) gives an approximation of $\max B(y)$ with additive error $O(\epsilon|\alpha|^r\|B\|_F)$ which is what we need (taking $A = \hat{B}$ of the previous subsection.)

5.4 Metric Tensors

Lemma 5.5. Let A be an r-dimensional tensor satisfying the following local density condition:

$$\forall i_1, \ldots, i_r \in V, \quad A_{i_1,\ldots,i_r} \leq \frac{c}{rn^{r-1}} \sum_{j=1}^{r} D_{i_j}$$

where c is a constant. Then A is a core-dense hypergraph with core-strength c.

Proof. We need to bound the core-strength of A. To this end,

$$\begin{aligned}
&\sum_{i_1,i_2,\ldots,i_r \in V} \frac{A^2_{i_1,\ldots,i_r}}{\prod_{j=1}^{r}(D_{i_j} + \bar{D})} \\
&\quad \leq \frac{c}{rn^{r-1}} \sum_{i_1,i_2,\ldots,i_r \in V} \frac{A_{i_1,\ldots,i_r} \sum_{j=1}^{r} D_{i_j}}{\prod_{j=1}^{r}(D_{i_j} + \bar{D})} \\
&\quad \leq \frac{c}{rn^{r-1}} \sum_{i_1,i_2,\ldots,i_r \in V} A_{i_1,\ldots,i_r} \sum_{j=1}^{r} \frac{1}{\prod_{k\in\{1,\ldots,r\}\setminus j}(D_{i_k} + \bar{D})} \\
&\quad \leq \frac{c}{rn^{r-1}} \left(\sum_{i_1,i_2,\ldots,i_r \in E} A_{i_1,\ldots,i_r} \right) \frac{r}{\bar{D}^{r-1}} \\
&\quad = \frac{c}{(\sum_{i=1}^{n} D_i)^{r-2}}.
\end{aligned}$$

Thus, the core-strength is at most

$$\left(\sum_{i=1}^{n} D_i \right)^{r-2} \sum_{i_1,i_2,\ldots,i_r \in E} \frac{A^2_{i_1,\ldots,i_r}}{\Pi_{j=1}^{r}(D_{i_j} + \bar{D})} \leq c. \qquad \square$$

Theorem 5.2 follows directly from Lemma 5.5 and Theorem 5.1. We next prove Corollary 5.3 for metrics.

Proof. (of Corollary 5.3) For $r = 2$, the condition of Theorem 5.2 says that for any $i, j \in V$,

$$A_{i,j} \leq \frac{c}{2n}(D_i + D_j).$$

We will verify that this holds for a metric MAX-2CSP with $c = 2$. When the entries of A form a metric, for any i, j, k, we have

$$A_{i,j} \leq A_{i,k} + A_{k,j}$$

and so

$$\begin{aligned} A_{i,j} &\leq \frac{1}{n}\left(\sum_{k=1}^{n} A_{i,k} + \sum_{k=1}^{n} A_{j,k}\right) \\ &= \frac{1}{n}(D_i + D_j). \end{aligned}$$

□

A non-negative real function d defined on $M \times M$ is called a *quasimetric* if $d(x, y) = 0$ when $x = y$, $d(x, y) = d(y, x)$ and $d(x, z) \leq C(d(x, y) + d(y, z))$, the last for some positive real number C, and all $x, y, z \in M$. Thus if it holds with $C = 1$, then d is a metric on M. The proof of Corollary 5.3 easily extends to quasi-metrics.

Quasi-metrics include a number of interesting distance functions which are not metrics, like the squares of Euclidean distances used in clustering applications.

5.5 Discussion

This chapter is based on Fernandez de la Vega et al. [25]. Prior to that paper, there was much progress on special cases. In particular, there were polynomial-time approximation schemes for *dense* unweighted problems [7, 24, 40, 47, 41, 4], and several cases of MAX-2CSP with metric weights including maxcut and partitioning [28, 50, 27, 26]. It is also shown in [25] that these methods can be applied to rCSPs with an additional constant number of global constraints, such as finding the maximum weight bisection.

Part II

Algorithms

6

Matrix Approximation via Random Sampling

In this chapter, we study randomized algorithms for matrix multiplication and low-rank 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.

6.1 Matrix–vector Product

In many numerical algorithms, a basic operation is the matrix–vector product. If A is an $m \times n$ 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 non-negative 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

$$\mathsf{Var}\,(X) = \mathsf{E}\,\|X\|^2 - \|\mathsf{E}\, X\|^2 = \sum_{j=1}^{n} \frac{\|A^{(j)}\|^2 v_j^2}{p_j} - \|Av\|^2. \tag{6.1}$$

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 (6.1) 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

$$\mathsf{Var}\, Y = \frac{1}{s}\sum_j \frac{\|A^{(j)}\|^2 v_j^2}{p_j} - \frac{1}{s}\|Av\|^2 \le \frac{1}{cs}\|A\|_F^2\|v\|^2. \tag{6.2}$$

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 $_{\mathrm{row}}$.

6.2 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 Equation (6.2) p times and adding, we get the next theorem (recall the notation that $B_{(j)}$ denotes row j of B).

Theorem 6.1. 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

$$Y = \frac{1}{s}\sum_{t=1}^{s} \frac{A^{(j_t)} B_{(j_t)}}{p_{j_t}},$$

$$\mathsf{E}\, Y = AB \quad \text{and} \quad \mathsf{Var}\, Y = \frac{1}{s}\sum_{j=1}^{n} \frac{\|A^{(j)}\|^2 \|B_{(j)}\|^2}{p_j} - \|AB\|_F^2. \tag{6.3}$$

If j_t are distributed according to $\mathrm{LS}_{\mathrm{col}}(A,c)$, then

$$\mathsf{Var}\, Y \le \frac{1}{cs}\|A\|_F^2\|B\|_F^2.$$

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 Equation (6.3)) 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. The next exercise establishes the optimality of the length-squared distribution.

Exercise 6.2. Suppose $a_1, a_2, \ldots, a_n$ are fixed positive reals. Prove that the minimum of the constrained optimization problem

$$\text{Min} \sum_{j=1}^{n} \frac{a_j}{x_j} \text{ subject to } x_j \geq 0\,;\, \sum_j x_j = 1$$

is attained at $x_j = \sqrt{a_j}/\sum_{i=1}^n \sqrt{a_i}$.

6.3 Low-Rank Approximation

When $B = A^T$, we may rewrite the expression (6.3) 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 6.3. 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,

$$\mathsf{E}\, CC^T = AA^T$$

and

$$\mathsf{E}\, \|CC^T - AA^T\|_F^2 = \frac{1}{s} \sum_{j=1}^{n} \frac{|A^{(j)}|^4}{p_j} - \frac{1}{s} \|AA^T\|_F^2.$$

If the p_js conform to the approximate length-squared distribution $\mathrm{LS}_{\mathrm{col}}(A, c)$, 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

$$\sum_t \left(\sigma_t(CC^T) - \sigma_t(AA^T) \right)^2 \leq \|CC^T - AA^T\|_F^2. \tag{6.4}$$

(Exercise 6.7 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 6.4. 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,

$$\|A - UU^T A\|_F^2 \leq \|A - A_k\|_F^2 + 2\sqrt{k}\|AA^T - CC^T\|_F$$
$$\|A - UU^T A\|_2^2 \leq \|A - A_k\|_2 + \|CC^T - AA^T\|_2 + \|CC^T - AA^T\|_F.$$

Proof. We have

$$\left\| A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A \right\|_F^2 = \|A\|_F^2 - \|U^T A\|_F^2$$

and

$$\|C_k\|_F^2 = \|U^T C\|_F^2.$$

Using these equations,

$$\begin{aligned}
&\|A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\|_F^2 - \|A - A_k\|_F^2 \\
&\quad = \|A\|_F^2 - \|U^T A\|_F^2 - (\|A\|_F^2 - \|A_k\|_F^2) \\
&\quad = \left(\|A_k\|_F^2 - \|C_k\|_F^2\right) + \|U^T C\|_F^2 - \|U^T A\|_F^2 \\
&\quad = \sum_{t=1}^{k} \left(\sigma_t(A)^2 - \sigma_t(C)^2\right) + \sum_{t=1}^{k} \left(\sigma_t(C)^2 - \left\|u^{(t)^T} A\right\|^2\right) \\
&\quad \le \sqrt{k \sum_{t=1}^{k} (\sigma_t(A)^2 - \sigma_t(C)^2)^2} + \sqrt{k \sum_{t=1}^{k} \left(\sigma_t(C)^2 - \left\|u^{(t)^T} A\right\|^2\right)^2} \\
&\quad = \sqrt{k \sum_{t=1}^{k} (\sigma_t(AA^T) - \sigma_t(CC^T))^2} \\
&\qquad + \sqrt{k \sum_{t=1}^{k} \left(u^{(t)^T} (CC^T - AA^T) u^{(t)}\right)^2} \\
&\quad \le 2\sqrt{k} \|AA^T - CC^T\|_F.
\end{aligned}$$

Here we first used the Cauchy–Schwarz inequality on both summations and then the Hoffman–Wielandt inequality (6.4).

The proof of the second statement also uses the Hoffman–Wielandt inequality. □

We can now combine Theorem 6.3 and Lemma 6.4 to obtain the main theorem of this section.

Theorem 6.5. Algorithm Fast SVD finds a rank-k matrix $\tilde{A}$ such that

$$\mathsf{E}\left(\|A - \tilde{A}\|_F^2\right) \le \|A - A_k\|_F^2 + 2\sqrt{\tfrac{k}{s}} \|A\|_F^2$$

$$\mathsf{E}\left(\|A - \tilde{A}\|_2^2\right) \le \|A - A_k\|_2 + \tfrac{2}{\sqrt{s}} \|A\|_F^2.$$

Exercise 6.6. Using the fact that $\|A\|_F^2 = \mathrm{Tr}(AA^T)$ show that:

1. For any two matrices P, Q, we have $|\mathrm{Tr}PQ| \leq \|P\|_F \|Q\|_F$.
2. For any matrix Y and any symmetric matrix X, $|\mathrm{Tr}XYX| \leq \|X\|_F^2 \|Y\|_F$.

Exercise 6.7. 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 6.8. (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 6.9. 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

$$\begin{aligned}
&\left\| A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A \right\|_F^2 - \left\| C - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} C \right\|_F^2 \\
&\quad \le \|A\|_F^2 - \|C\|_F^2 + \|AA^T - CC^T\|_F \left\| \sum_{t=1}^{k} u^{(t)} u^{(t)^T} \right\|_F \\
&\quad\quad \times \left(2 + \left\| \sum_{t=1}^{k} u^{(t)} u^{(t)^T} \right\|_F \right) \left\| A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A \right\|_2^2 \qquad (6.5) \\
&\quad\quad - \left\| C - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} C \right\|_2^2 \\
&\quad \le \|AA^T - CC^T\|_2 \left(\left\| \sum_{t=1}^{k} u^{(t)} u^{(t)^T} \right\|_2 + 1 \right)^2 . \qquad (6.6)
\end{aligned}$$

Proof.

$$\begin{aligned}
&\left\| A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A \right\|_F^2 \\
&\quad = \mathrm{Tr} \left(\left(A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A \right) \left(A^T - A^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} \right) \right) \\
&\quad = \mathrm{Tr} AA^T + \mathrm{Tr} \sum_{t=1}^{k} u^{(t)} u^{(t)^T} AA^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} - 2\mathrm{Tr} \sum_{t=1}^{k} u^{(t)} u^{(t)^T} AA^T,
\end{aligned}$$

where we have used the fact that square matrices commute under trace. We do the same expansion for C to get

$$\begin{aligned}
&\left\|A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\right\|_F^2 - \left\|C - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} C\right\|_F^2 - \left(\|A\|_F^2 - \|C\|_F^2\right) \\
&\quad = \mathrm{Tr} \sum_{t=1}^{k} u^{(t)} u^{(t)^T} (AA^T - CC^T) \sum_{t=1}^{k} u^{(t)} u^{(t)^T} \\
&\qquad - 2\mathrm{Tr} \sum_{t=1}^{k} u^{(t)} u^{(t)^T} (AA^T - CC^T) \\
&\quad \leq \left\|\sum_{t=1}^{k} u^{(t)} u^{(t)^T}\right\|_F^2 \|AA^T - CC^T\|_F \\
&\qquad + 2\left\|\sum_{t=1}^{k} u^{(t)} u^{(t)^T}\right\|_F \|AA^T - CC^T\|_F,
\end{aligned}$$

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 6.6). This gives us Equation (6.5).

For Equation (6.6), suppose v is the unit length vector achieving

$$\left\|v^T \left(A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\right)\right\| = \left\|A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\right\|_2.$$

Then we expand

$$\begin{aligned}
&\left\|v^T \left(A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\right)\right\|^2 \\
&\quad = v^T \left(A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A\right) \left(A^T - A^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T}\right) v \\
&\quad = v^T AA^T v - 2v^T AA^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} v \\
&\qquad + v^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} AA^T \sum_{t=1}^{k} u^{(t)} u^{(t)^T} v,
\end{aligned}$$

and the corresponding terms for C. Now, Equation (6.6) follows by a somewhat tedious but routine calculation. □

6.4 Invariant Subspaces

The classical SVD has associated with it the decomposition of space into the **direct sum** of **invariant subspaces**.

Theorem 6.10. Let A be an $m \times n$ matrix and $v^{(1)}, v^{(2)}, \ldots, v^{(n)}$ an orthonormal basis for $\mathbf{R}^n$. Suppose for $k, 1 \leq k \leq \text{rank}(A)$ 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:

$$\sum_{t=1}^{k} |u^{(t)^T} A|^2 = \sum_{t=1}^{k} \sigma_t^2$$

$$\left\| A - A\sum_{t=1}^{k} v^{(t)} v^{(t)^T} \right\|_F^2 = \left\| A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A \right\|_F^2 = \sum_{t=k+1}^{n} \sigma_t^2(A)$$

$$\left\| A - A\sum_{t=1}^{k} v^{(t)} v^{(t)^T} \right\|_2 = \left\| A - \sum_{t=1}^{k} u^{(t)} u^{(t)^T} A \right\|_2 = \sigma_{k+1}(A).$$

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 6.11. Prove Theorem 6.10.

6.4.1 Approximate Invariance

The theorem below proves that 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

$$\Delta(A, v^{(1)}, v^{(2)}, \ldots, v^{(k)}) = \text{Max}_{1 \leq t \leq k} \sum_{i=1}^{t} (\sigma_i^2(A) - |Av^{(i)}|^2) \qquad (6.7)$$

Theorem 6.12. 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 $t, 1 \leq t \leq r$, we have

$$\sum_{s=t+1}^{r} \left(v^{(t)^T} A^T A v^{(s)} \right)^2 \leq |Av^{(t)}|^2 \big(\sigma_1^2(A) + \sigma_2^2(A) + \ldots \sigma_t^2(A) - |Av^{(1)}|^2 - |Av^{(2)}|^2 - \ldots |Av^{(t)}|^2 \big).$$

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 6.12 implies the classical Theorem 6.10: $\sigma_t(A) = |Av^{(t)}|$ implies that the right-hand side of the inequality above is zero. Thus, $v^{(t)^T} A^T A$ is collinear 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

$$\begin{aligned}\sum_{s=2}^{r}\left(v^{(1)^T}A^TAv^{(s)}\right)^2 &= |v^{(1)^T}A^TA|^2 - \left(v^{(1)^T}A^TAv^{(1)}\right)^2\\ &\le |Av^{(1)}|^2\sigma_1(A)^2 - |Av^{(1)}|^4\\ &\le |Av^{(1)}|^2\left(\sigma_1(A)^2 - |Av^{(1)}|^2\right). \qquad (6.8)\end{aligned}$$

The proof of the theorem will be by induction on the rank of A. If $r = 1$, there is nothing to prove. Assume $r \ge 2$. 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, $1 \le s \le r-1$,

$$\begin{aligned}&\sigma_1^2(A') + \sigma_2^2(A') + \ldots\sigma_s^2(A')\\ &\quad= \sum_{t=1}^{s}|A'w^{(t)}|^2\\ &\quad= \sum_{t=1}^{s}|Aw^{(t)}|^2\\ &\quad= |Av^{(1)}|^2 + \sum_{t=1}^{s}|Aw^{(t)}|^2 - |Av^{(1)}|^2\\ &\quad\le \mathrm{MAX}_{\substack{u^{(1)},u^{(2)}\ldots u^{(s+1)}\\ \text{orthonormal}}}\sum_{t=1}^{s+1}|Au^{(t)}|^2 - |Av^{(1)}|^2\\ &\quad= \sigma_1(A)^2 + \sigma_2(A)^2 + \ldots\sigma_{s+1}(A)^2 - |Av^{(1)}|^2, \qquad (6.9)\end{aligned}$$

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 $t, 2 \le t \le r$,

$$\begin{aligned}\sum_{s=t+1}^{r}(v^{(t)^T}A'^TA'v^{(s)})^2 \le |A'v^{(t)}|^2\big(&\sigma_1^2(A') + \sigma_2^2(A') + \ldots\sigma_{t-1}^2(A')\\ &- |A'v^{(2)}|^2 - |A'v^{(3)}|^2 - \ldots|A'v^{(t)}|^2\big)\end{aligned}$$

Note that for $t \geq 2$, we have $A'v^{(t)} = Av^{(t)}$. So, we get using Equation (6.9)

$$\sum_{s=t+1}^{r} ({v^{(t)}}^T A^T A v^{(s)})^2 \leq |Av^{(t)}|^2 \big(\sigma_1^2(A) + \sigma_2^2(A) + \ldots \sigma_t^2(A) - |Av^{(1)}|^2 - |Av^{(2)}|^2 - \ldots |Av^{(t)}|^2\big).$$

This together with Equation (6.8) finishes the proof of the Theorem. □

We will use Theorem 6.12 to prove Theorem 6.13 below. Theorem 6.13 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 6.13 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 6.10 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 6.14 below) to get a reasonable bound.

Theorem 6.13. 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 Equation (6.7). Let

$$u^{(t)} = \frac{Av^{(t)}}{|Av^{(t)}|} \quad \text{for } t = 1, 2, \ldots, k.$$

Then

$$\left\| \sum_{t=1}^{k} u^{(t)} {u^{(t)}}^T A - A \right\|_F^2 \leq \left\| A - \sum_{t=1}^{k} Av^{(t)} {v^{(t)}}^T \right\|_F^2 + \left(\sum_{t=1}^{k} \frac{2}{|Av^{(t)}|^2} \right) \left(\sum_{t=1}^{k} |Av^{(t)}|^2 \right) \Delta$$

$$\left\|\sum_{t=1}^{k} u^{(t)}u^{(t)^T}A - A\right\|_2^2 \le \left\|A - \sum_{t=1}^{k} Av^{(t)}v^{(t)^T}\right\|_2^2 + \left(\sum_{t=1}^{k}\frac{2}{|Av^{(t)}|^2}\right)\left(\sum_{t=1}^{k}|Av^{(t)}|^2\right)\Delta.$$

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

$$w^{(t)^T} = v^{(t)^T}A^TA - |Av^{(t)}|^2 v^{(t)^T}$$

be the component of $v^{(t)^T}A^TA$ orthogonal to $v^{(t)^T}$. We have

$$u^{(t)}u^{(t)^T}A = \frac{Av^{(t)}v^{(t)^T}A^TA}{|Av^{(t)}|^2} = Av^{(t)}v^{(t)^T} + Av^{(t)}w^{(t)^T}.$$

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

$$\begin{aligned}
&\left\|\sum_t u^{(t)}u^{(t)^T}A - A\right\|_F^2 \\
&\quad= \left\|\sum_t Av^{(t)}v^{(t)^T} + \sum_t \frac{Av^{(t)}w^{(t)^T}}{|Av^{(t)}|^2} - A\right\|_F^2 \\
&\quad= \left\|A - \sum_t Av^{(t)}v^{(t)^T}\right\|_F^2 + \left(\sum_t \left|\frac{Av^{(t)}}{|Av^{(t)}|^2}\right|\left|w^{(t)}\right|\right)^2 \\
&\qquad -2\sum_{s=1}^{r}\sum_t \left(v^{(s)^T}w^{(t)}\right)\frac{v^{(t)^T}A^T}{|Av^{(t)}|^2}\left(A - \sum_t Av^{(t)}v^{(t)^T}\right)v^{(s)} \\
&\quad\le \left\|A - \sum_t Av^{(t)}v^{(t)^T}\right\|_F^2 + \left(\sum_t |w^{(t)}|^2\right)\left(\sum_t \frac{1}{|Av^{(t)}|^2}\right) \\
&\qquad -2\sum_{s=k+1}^{r}\sum_t \frac{(v^{(t)^T}A^TAv^{(s)})^2}{|Av^{(t)}|^2}
\end{aligned}$$

$$\begin{aligned}
&\text{since } \left(A - \textstyle\sum_t Av^{(t)}v^{(t)^T}\right)v^{(s)} = 0 \quad \text{for } s \le k \\
&\text{and } v^{(s)^T}w^{(t)} = v^{(s)^T}A^TAv^{(t)} \\
&\le \left\|A - \sum_t Av^{(t)}v^{(t)^T}\right\|_F^2 \\
&\quad + \left(\sum_t \frac{1}{|Av^{(t)}|^2}\right)\left(2\sum_t \sum_{s=t+1}^{r}\left(v^{(t)^T}A^TAv^{(s)}\right)^2\right) \\
&\le \left\|A - \sum_t Av^{(t)}v^{(t)^T}\right\|_F^2 + \left(\sum_t \frac{2}{|Av^{(t)}|^2}\right)\left(\sum_t |Av^{(t)}|^2\right)\Delta,
\end{aligned}$$

using Theorem 6.12.

For the 2-norm, the argument is similar. Suppose a vector p achieves

$$\left\|\sum_t u^{(t)}u^{(t)^T}A - A\right\|_2 = \left|\left(\sum_t u^{(t)}u^{(t)^T}A - A\right)p\right|.$$

We now use

$$|(X+Y)p|^2 = p^TX^TXp + p^TY^TYp + 2p^TX^TYp$$

to get

$$\begin{aligned}
&\left\|\sum_t u^{(t)}u^{(t)^T}A - A\right\|_2^2 \\
&\quad \le \left\|A - \sum_t Av^{(t)}v^{(t)^T}\right\|_2^2 + \left(\sum_t |w^{(t)}|^2\right)\left(\sum_t \frac{1}{|Av^{(t)}|^2}\right) \\
&\quad\quad -2\sum_t (p^Tw^{(t)})\frac{v^{(t)^T}A^T}{|Av^{(t)}|^2}\left(A - \sum_t Av^{(t)}v^{(t)^T}\right)p.
\end{aligned}$$

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

$$\sum_t \left(p^T w^{(t)}\right) \frac{v^{(t)^T} A^T}{|Av^{(t)}|^2} \left(A - \sum_t Av^{(t)} v^{(t)^T}\right) p$$
$$= \sum_t \left(p^{(2)^T} w^{(t)}\right) \frac{v^{(t)^T} A^T}{|Av^{(t)}|^2} Ap^{(2)} = \frac{\sum_t \left(v^{(t)^T} A^T Ap^{(2)}\right)^2}{|Av^{(t)}|^2},$$

where we have used the fact that $p^{(2)}$ is orthogonal to $v^{(t)}$ to get $p^{(2)^T} w^{(t)} = v^{(t)^T} A^T Ap^{(2)}$. □

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 6.14. 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

$$u^{(t)} = \frac{Av^{(t)}}{|Av^{(t)}|} \quad \text{for } t = 1, 2, \ldots, k.$$

Define l to be the largest integer in $\{1, 2, \ldots, k\}$ such that $|Av^{(l)}|^2 \geq \sqrt{\delta}\|A\|_F^2$. Then,

$$\mathsf{E}\left\|A - \sum_{t=1}^{l} u^{(t)} u^{(t)^T} A\right\|_F^2 \leq \mathsf{E}\left\|A - A\sum_{t=1}^{k} v^{(t)} v^{(t)^T}\right\|_F^2 + 3k\sqrt{\delta}\|A\|_F^2.$$

$$\mathsf{E}\left\|A - \sum_{t=1}^{l} u^{(t)} u^{(t)^T} A\right\|_2^2 \leq \mathsf{E}\left\|A - A\sum_{t=1}^{k} v^{(t)} v^{(t)^T}\right\|_2^2 + 3k\sqrt{\delta}\|A\|_F^2$$

Proof. We apply the theorem with k replaced by l and taking expectations of both sides (which are now random variables) to get

$$\begin{aligned}
&\mathsf{E}\left\|A-\sum_{t=1}^{l}u^{(t)}u^{(t)^T}\right\|_F^2 \\
&\quad\le \mathsf{E}\left\|A-A\sum_{t=1}^{l}v^{(t)}v^{(t)^T}\right\|_F^2+\frac{2k}{\sqrt{\delta}}\mathsf{E}\left(\sum_{t=1}^{l}\left(\sigma_t^2(A)-|Av^{(t)}|^2\right)\right) \\
&\quad\le \mathsf{E}\left\|A-A\sum_{t=1}^{k}v^{(t)}v^{(t)^T}\right\|_F^2+\sum_{t=l+1}^{k}|Av^{(t)}|^2+2k\sqrt{\delta}\|A\|_F^2,
\end{aligned}$$

where, we have used the fact that from the minimax principle and $|Av^{(1)}|\ge|Av^{(2)}|\ge\ldots|Av^{(k)}|>0$, we get that $\sigma_t(A)\ge|Av^{(t)}|$ 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

$$\left\|A-A\sum_{t=1}^{l}v^{(t)}v^{(t)^T}\right\|_2^2\le\left\|A-A\sum_{t=1}^{k}v^{(t)}v^{(t)^T}\right\|_2^2+\sum_{t=l+1}^{k}|Av^{(t)}|^2.$$

To see this, if p is the top left singular vector of $A-A\sum_{t=1}^{l}v^{(t)}v^{(t)^T}$, then

$$\begin{aligned}
\left|p^T\left(A-A\sum_{t=1}^{l}v^{(t)}v^{(t)^T}\right)\right|^2 &= p^TAA^Tp-p^TA\sum_{t=1}^{l}v^{(t)}v^{(t)^T}A^Tp \\
&\le \left\|A-A\sum_{t=1}^{k}v^{(t)}v^{(t)^T}\right\|_2^2+\sum_{t=l+1}^{k}|p^TAv^{(t)}|^2.
\end{aligned}$$

□

6.5 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

$$s=\frac{c_1k^5}{c\epsilon^4}$$

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 an $s \times s$ matrix W. By Theorem 6.3, we have

$$\mathsf{E}\,\|C^TC - W^TW\|_F \leq \frac{1}{\sqrt{cs}}\mathsf{E}\,\|C\|_F^2 = \frac{c_2\epsilon^2}{k^{2.5}}\|A\|_F^2, \tag{6.10}$$

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^TW (note : This is just an $s \times s$ matrix !) say

$$W^TW = \sum_t \sigma_t^2(W) v^{(t)} v^{(t)^T}.$$

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 6.4 with C replacing A and W replacing C in that Lemma and right projections instead of left projections. Hence we get (using Equation (6.10))

$$\mathsf{E}\left\|C - C\sum_{t=1}^k v^{(t)}v^{(t)^T}\right\|_F^2 \leq \mathsf{E}\,\|C\|_F^2 - \mathsf{E}\sum_{t=1}^k \sigma_t^2(C) + \frac{c_3\epsilon^2}{k^2}\|A\|_F^2 \tag{6.11}$$

$$\mathsf{E}\left\|C - C\sum_{t=1}^k v^{(t)}v^{(t)^T}\right\|_2^2 \leq \mathsf{E}\,\sigma_{k+1}(C)^2 + (2+4k)O\left(\frac{\epsilon^2}{k^3}\right)\mathsf{E}\,\|C\|_F^2 \tag{6.12}$$

$$\leq \sigma_{k+1}^2(A) + \frac{c_4\epsilon^2}{k^2}\|A\|_F^2. \tag{6.13}$$

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 Equation (6.13)

$$\mathsf{E}\sum_{t=1}^k \left(\sigma_t^2(C) - |Cv^{(t)}|^2\right) \leq \frac{c_5\epsilon^2}{k^2}\|A\|_F^2. \tag{6.14}$$

Equation (6.13) also yields

$$\mathsf{E}\left\|C - C\sum_{t=1}^{k} v^{(t)}v^{(t)^T}\right\|_F^2 \leq \|A\|_F^2 - \sum_{t=1}^{k}\sigma_t^2(A) + \|A\|_F^2\frac{c_6\epsilon^2}{k^2}$$

$$\text{Thus,}\quad \mathsf{E}\left\|C - C\sum_{t=1}^{k} v^{(t)}v^{(t)^T}\right\|_F^2 \leq \sum_{t=k+1}^{n}\sigma_t^2(A) + \frac{c_6\epsilon^2}{k^2}\|A\|_F^2 \qquad (6.15)$$

Now we wish to use Corollary 6.14 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

$$\begin{aligned}\mathsf{E}\left\|C - \sum_{t=1}^{l} u^{(t)}u^{(t)^T}C\right\|_F^2 &\leq \mathsf{E}\left\|C - C\sum_{t=1}^{k} v^{(t)}v^{(t)^T}\right\|_F^2 + O(\epsilon)\|A\|_F^2 \\ &\leq \sum_{t=k+1}^{n}\sigma_t^2(A) + O(\epsilon)\|A\|_F^2. \qquad (6.16)\end{aligned}$$

$$\mathsf{E}\left\|C - \sum_{t=1}^{l} u^{(t)}u^{(t)^T}C\right\|_2^2 \leq \sigma_{k+1}^2(A) + O(\epsilon)\|A\|_F^2. \qquad (6.17)$$

Finally, we use Lemma 6.9 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,

$$\begin{aligned}\mathsf{E}\left\|A - \sum_{t=1}^{l} u^{(t)}u^{(t)^T}A\right\|_F^2 &\leq \mathsf{E}\left\|C - \sum_{t=1}^{l} u^{(t)}u^{(t)^T}C\right\|_F^2 \\ &\quad + \frac{1}{\sqrt{cs}}\|A\|_F^2 k(2+k) \\ &\leq \sum_{t=k+1}^{n}\sigma_t^2(A) + O(\epsilon)\|A\|_F^2\end{aligned}$$

$$\mathsf{E}\left\|A - \sum_{t=1}^{l} u^{(t)}u^{(t)^T}A\right\|_2^2 \leq \sigma_{k+1}^2(A) + O(\epsilon)\|A\|_F^2.$$

Thus, we get the following lemma:

Lemma 6.15. 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:

$$\mathsf{E}\left\|A - \sum_{t=1}^{l} u^{(t)} u^{(t)^T} A\right\|_F^2 \leq \sum_{t=k+1}^{n} \sigma_t^2(A) + \epsilon \|A\|_F^2$$

$$\mathsf{E}\left\|A - \sum_{t=1}^{l} u^{(t)} u^{(t)^T} A\right\|_2^2 \leq \sigma_{k+1}^2(A) + \epsilon \|A\|_F^2.$$

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 an $\mathrm{LS}_{\mathtt{row}}(C,c)$ distribution and scale to form an $s \times s$ 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
$$u^{(t)} = \frac{Cv^{(t)}}{|Cv^{(t)}|} \quad \text{for } t = 1,2,\ldots,k.$$

Let l to be the largest integer in $\{1, 2, \ldots, k\}$ such that

$$|Cv^{(l)}|^2 \geq c_9 \epsilon \|C\|_F^2 / k.$$

5. Return

$$\sum_{t=1}^{l} u^{(t)} u^{(t)^T} A$$

as the approximation to A.

6.6 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 an $m \times n$ matrix and R (R for rows) is an $s \times n$ matrix constructed 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 an $m \times s$ 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 $R^T R \approx A^T A$. Now it is easy to see that if we are given both AA^T and $A^T A$ 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

$$AA^T = \sum_t \sigma_t^2(A) u^{(t)} u^{(t)^T} \quad A^T A = \sum_t \sigma_t^2(A) v^{(t)} v^{(t)^T},$$

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, $AA^T = A^T A = I$.] 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 6.16. 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 $\mathrm{LS}_{\mathrm{rows}(A,c)}$ 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

$$\mathsf{E}\left(\|CUR - A\|_F\right) \leq \|A - A_k\|_F + \sqrt{\tfrac{k}{cs}}\|A\|_F + \sqrt{2}k^{\frac{1}{4}}\|AA^T - CC^T\|_F^{1/2}$$

$$\mathsf{E}\left(\|CUR - A\|_2\right) \leq \|A - A_k\|_2 + \sqrt{\tfrac{k}{cs}}\|A\|_F + \sqrt{2}\|AA^T - CC^T\|_F^{1/2}$$

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 6.1 directly, we thus get using $\| \sum_{t=1}^{k} x^{(t)} x^{(t)^T} \|_F^2 = k$. (Note: 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.)

$$\mathsf{E} \left\| \sum_{t=1}^{k} x^{(t)} x^{(t)^T} D^T DA - \sum_{t=1}^{k} x^{(t)} x^{(t)^T} A \right\|_F^2 \leq \frac{k}{cs} \|A\|_F^2.$$

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 6.4 it follows that

$$\left\|\sum_{t=1}^{k} x^{(t)} x^{(t)^T} A - A\right\|_F^2 \leq \|A - A_k\|_F^2 + 2\sqrt{k}\|AA^T - CC^T\|_F$$
$$\left\|\sum_{t=1}^{k} x^{(t)} x^{(t)^T} A - A\right\|_2^2 \leq \|A - A_k\|_2 + 2\|AA^T - CC^T\|_F.$$

Now the theorem follows using the triangle inequality on the norms. □

As a corollary, we have the following:

Corollary 6.17. Suppose we are given C, a set of independently chosen columns of A from $\mathrm{LS}_{col(A,c)}$ and R, a set of s independently chosen rows of A from $\mathrm{LS}_{rows(A,c)}$. 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 an $s \times s$ matrix U such that

$$\|A - CUR\|_F \leq \mathrm{poly}\left(\frac{\delta}{s}\right)\|A\|_F?$$

So we do not assume that R is a random sample as in the theorem.

6.7 Discussion

Sampling from the length square distribution was introduced in a paper by Frieze et al. [42, 43] 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 [31]. The linear-time low-rank approximation was given by Drineas et al. [33]. The CUR decomposition first appeared in [32]. The best-know sample complexity for the constant-time algorithm is $O(k^2/\epsilon^4)$ and other refinements are given in [34, 35, 36]. An alternative sampling method which sparsifies a given matrix and uses a low-rank approximation of the sparse matrix was given in [2].

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 booksellers, movie renters, etc.

In the first mathematical model for Recommendation Systems Azar et al. [10] 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 [37].

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.

7

Adaptive Sampling Methods

In this chapter, we continue our study of sampling methods for matrix approximation, including linear regression and low-rank approximation. In the previous chapter, we saw that any matrix A has a subset of k/ϵ rows whose span contains an approximately optimal rank-k approximation to A. We recall the precise statement.

Theorem 7.1. Let S be a sample of s rows of an $m \times n$ matrix A, each chosen independently from the following distribution: Row i is picked with probability

$$P_i \geq c\frac{\|A^{(i)}\|^2}{\|A\|_F^2}.$$

If $s \geq k/c\epsilon$, then the span of S contains a matrix $\tilde{A}_k$ of rank at most k for which

$$\mathsf{E}\left(\|A - \tilde{A}_k\|_F^2\right) \leq \|A - A_k\|_F^2 + \epsilon\|A\|_F^2.$$

This was turned into an efficient algorithm. The algorithm makes one pass through A to figure out the sampling distribution

and another pass to compute the approximation. Its complexity is $O(\min\{m,n\}k^2/\epsilon^4)$. We also saw a "constant-time" algorithm that samples both rows and columns.

These results naturally lead to the following two important questions: (1) The additive error in Theorem 7.1 is $\epsilon\|A\|_F^2$ which can be very large since we have no control on $\|A\|_F^2$. Can this error be reduced significantly by using multiple passes through the data? (2) Can we get multiplicative $(1+\epsilon)$ approximations using a small sample?

7.1 Adaptive Length-Squared Sampling

As an illustrative example, suppose the data consists of points along a one-dimensional subspace of $\boldsymbol{R}^n$ except for one point. The best rank-2 subspace has zero error. However, one round of sampling will most likely miss the point far from the line. So we use a two-round approach. In the first pass, we get a sample from the squared length distribution and find a rank-2 subspace using it. Then we sample again, but this time with probability proportional to the squared distance to the first subspace. If the lone far-off point is missed in the first pass, it will have a high probability of being chosen in the second pass. The span of the full sample now contains a good rank-2 approximation.

The main idea behind the adaptive length-squared sampling scheme is the following generalization of Theorem 7.1. Notice that if we put $V = \emptyset$ in the following theorem then we get exactly Theorem 7.1. Recall that for a subspace $V \subseteq \boldsymbol{R}^n$, we denote by $\pi_{V,k}(A)$ the best rank-k approximation (under the Frobenius norm) of A with rows in the span of V.

Theorem 7.2. Let $A \in \boldsymbol{R}^{m\times n}$. Let $V \subseteq \boldsymbol{R}^n$ be a vector subspace. Let $E = A - \pi_V(A)$. For a fixed $c \in \boldsymbol{R}$, let S be a random sample of s rows of A from a distribution such that row i is chosen with probability

$$P_i \geq c\frac{\|E^{(i)}\|^2}{\|E\|_F^2}. \tag{7.1}$$

Then, for any non-negative integer k,

$$\mathsf{E}_S(\|A - \pi_{V+\mathrm{span}(S),k}(A)\|_F^2) \leq \|A - \pi_k(A)\|_F^2 + \frac{k}{cs}\|E\|_F^2.$$

Proof. For $S = (r_i)_{i=1}^s$ a sample of rows of A and $1 \le j \le r$, let

$$w^{(j)} = \pi_V(A)^T u^{(j)} + \frac{1}{s}\sum_{i=1}^{s} \frac{u_{r_i}^{(j)}}{P_{r_i}} E^{(r_i)}.$$

Then, $\mathsf{E}_S(w^{(j)}) = \pi_V(A)^T u^{(j)} + E^T u^{(j)} = \sigma_j v^{(j)}$. Now we will bound $\mathsf{E}_S(\|w^{(j)} - \sigma_j v^{(j)}\|^2)$. Use the definition of $w^{(j)}$ to get

$$w^{(j)} - \sigma_j v^{(j)} = \frac{1}{s}\sum_{i=1}^{s} \frac{u_{r_i}^{(j)}}{P_{r_i}} E^{(r_i)} - E^T u^{(j)}.$$

Apply the norm squared to each side and expand the left-hand side:

$$\begin{aligned}\|w^{(j)} - \sigma_j v^{(j)}\|^2 = &\left\| \frac{1}{s}\sum_{i=1}^{s} \frac{u_{r_i}^{(j)}}{P_{r_i}} E^{(r_i)} \right\|^2 - \frac{2}{s}\sum_{i=1}^{s} \frac{u_{r_i}^{(j)}}{P_{r_i}} E^{(r_i)} \cdot (E^T u^{(j)}) \\ &+ \|E^T u^{(j)}\|^2. \end{aligned} \tag{7.2}$$

Observe that

$$\mathsf{E}_S\left(\frac{u_{r_i}^{(j)}}{P_{r_i}} E^{(r_i)}\right) = \sum_{i=1}^{m} P_i \frac{u_i^{(j)}}{P_i} E^{(i)} = E^T u^{(j)}, \tag{7.3}$$

which implies that

$$\mathsf{E}_S\left(\frac{2}{s}\sum_{i=1}^{s} \frac{u_{r_i}^{(j)}}{P_{r_i}} E^{(r_i)} \cdot (E^T u^{(j)})\right) = 2\|E^T u^{(j)}\|^2.$$

Using this, apply E_S to Equation (7.2) to get:

$$\mathsf{E}_S(\|w^{(j)} - \sigma_j v^{(j)}\|^2) = \mathsf{E}_S\left(\left\| \frac{1}{s}\sum_{i=1}^{s} \frac{u_{r_i}^{(j)}}{P_{r_i}} E^{(r_i)} \right\|^2\right) - \|E^T u^{(j)}\|^2 \tag{7.4}$$

Now, from the left-hand side, and expanding the norm squared,

$$\begin{aligned}&\mathsf{E}_S\left(\left\| \frac{1}{s}\sum_{i=1}^{s} \frac{u_{r_i}^{(j)}}{P_{r_i}} E^{(r_i)} \right\|^2\right) \\ &\quad = \frac{1}{s^2}\sum_{i=1}^{s} \mathsf{E}_S\left(\frac{\|u_{r_i}^{(j)} E^{(r_i)}\|^2}{P_{r_i}^2}\right) + \frac{2}{s^2}\sum_{1\le i<l\le s} \mathsf{E}_S\left(\frac{u_{r_i}^{(j)} E^{(r_i)}}{P_{r_i}} \cdot \frac{u_{r_l}^{(j)} E^{(r_l)}}{P_{r_l}}\right)\end{aligned} \tag{7.5}$$

where

$$\sum_{i=1}^{s} \mathsf{E}_S\left(\frac{\|u_{r_i}^{(j)}E^{(r_i)}\|^2}{P_{r_i}^2}\right) = \sum_{i=1}^{s}\sum_{l=1}^{m} P_l \frac{\|u_l^{(j)}E^{(l)}\|^2}{P_l^2} = s\sum_{l=1}^{m}\frac{\|u_l^{(j)}E^{(l)}\|^2}{P_l} \tag{7.6}$$

and, using the independence of the r_is and Equation (7.3),

$$\begin{aligned}
&\sum_{1\le i<l\le s} \mathsf{E}_S\left(\frac{u_{r_i}^{(j)}E^{(r_i)}}{P_{r_i}}\cdot\frac{u_{r_l}^{(j)}E^{(r_l)}}{P_{r_l}}\right)\\
&\quad= \sum_{1\le i<l\le s} \mathsf{E}_S\left(\frac{u_{r_i}^{(j)}E^{(r_i)}}{P_{r_i}}\right)\cdot \mathsf{E}_S\left(\frac{u_{r_l}^{(j)}E^{(r_l)}}{P_{r_l}}\right)\\
&\quad= \frac{s(s-1)}{2}\|E^T u^{(j)}\|^2.
\end{aligned} \tag{7.7}$$

The substitution of Equations (7.6) and (7.7) in Equation (7.5) gives

$$\mathsf{E}_S\left(\left\|\frac{1}{s}\sum_{i=1}^{s}\frac{u_{r_i}^{(j)}}{P_{r_i}}E^{(r_i)}\right\|^2\right) = \frac{1}{s}\sum_{i=1}^{m}\frac{\|u_i^{(j)}E^{(i)}\|^2}{P_i} + \frac{s-1}{s}\|E^T u^{(j)}\|^2.$$

Using this in Equation (7.4) we have

$$\mathsf{E}_S(\|w^{(j)} - \sigma_j v^{(j)}\|^2) = \frac{1}{s}\sum_{i=1}^{m}\frac{\|u_i^{(j)}E^{(i)}\|^2}{P_i} - \frac{1}{s}\|E^T u^{(j)}\|^2,$$

and, using the hypothesis for P_i (Equation (7.1)), remembering that $u^{(j)}$ is a unit vector and discarding the second term we conclude

$$\mathsf{E}_S(\|w^{(j)} - \sigma_j v^{(j)}\|^2) \le \frac{1}{cs}\|E\|_F^2. \tag{7.8}$$

Let $\hat{y}^{(j)} = \frac{1}{\sigma_j}w^{(j)}$ for $j = 1,\ldots,r$, let $k' = \min\{k,r\}$ (think of k' as equal to k, this is the interesting case), let $W = \text{span}\{\hat{y}^{(1)},\ldots,\hat{y}^{(k')}\}$, and $\hat{F} = A\sum_{t=1}^{k'} v^{(t)}\hat{y}^{(t)T}$. We will bound the error $\|A - \pi_W(A)\|_F^2$ using $\hat{F}$. Observe that the row space of $\hat{F}$ is contained in W and π_W is the projection operator onto the subspace of all matrices with row space in W with respect to the Frobenius norm. Thus,

$$\|A - \pi_W(A)\|_F^2 \le \|A - \hat{F}\|_F^2. \tag{7.9}$$

Moreover,

$$\|A - \hat{F}\|_F^2 = \sum_{i=1}^{r} \|(A - \hat{F})^T u^{(i)}\|^2 = \sum_{i=1}^{k'} \|\sigma_i v^{(i)} - w^{(i)}\|^2 + \sum_{i=k'+1}^{r} \sigma_i^2. \tag{7.10}$$

Taking expectation and using Equation (7.8) we get

$$\mathsf{E}_S(\|A - \hat{F}\|_F^2) \leq \sum_{i=k+1}^{n} \sigma_i^2 + \frac{k}{cs}\|E\|_F^2 = \|A - \pi_k(A)\|_F^2 + \frac{k}{cs}\|E\|_F^2.$$

This and Equation (7.9) give

$$\mathsf{E}_S(\|A - \pi_W(A)\|_F^2) \leq \|A - \pi_k(A)\|_F^2 + \frac{k}{cs}\|E\|_F^2. \tag{7.11}$$

Finally, the fact that $W \subseteq V + \text{span}(S)$ and $\dim(W) \leq k$ imply that

$$\|A - \pi_{V+\text{span}(S),k}(A)\|_F^2 \leq \|A - \pi_W(A)\|_F^2,$$

and, combining this with Equation (7.11), we conclude

$$\mathsf{E}_S(\|A - \pi_{V+\text{span}(S),k}(A)\|_F^2) \leq \|A - \pi_k(A)\|_F^2 + \frac{k}{cs}\|E\|_F^2.$$

□

Now we can use Theorem 7.2 to prove the main theorem of this section by induction.

Theorem 7.3. Let $S = S_1 \cup \cdots \cup S_t$ be a random sample of rows of an $m \times n$ matrix A, where for $j = 1, \ldots, t$, each set S_j is a sample of s rows of A chosen independently from the following distribution: row i is picked with probability

$$P_i^{(j)} \geq c\frac{\|E_j^{(i)}\|^2}{\|E_j\|_F^2}$$

where $E_1 = A$, $E_j = A - \pi_{S_1 \cup \cdots \cup S_{j-1}}(A)$ and c is a constant. Then for $s \geq k/c\epsilon$, the span of S contains a matrix $\tilde{A}_k$ of rank k such that

$$\mathsf{E}_S(\|A - \tilde{A}_k\|_F^2) \leq \frac{1}{1-\epsilon}\|A - A_k\|_F^2 + \epsilon^t\|A\|_F^2.$$

Proof. We will prove the slightly stronger result

$$\mathsf{E}_S(\|A - \pi_{S,k}(A)\|_F^2) \leq \frac{1 - (\frac{k}{cs})^t}{1 - \frac{k}{cs}} \|A - \pi_k(A)\|_F^2 + \left(\frac{k}{cs}\right)^t \|A\|_F^2$$

by induction on t. The case $t = 1$ is precisely Theorem 7.1.

For the inductive step, let $E = A - \pi_{S_1 \cup \cdots \cup S_{t-1}}(A)$. By means of Theorem 7.2 we have that,

$$\mathsf{E}_{S_t}(\|A - \pi_{S_1 \cup \cdots \cup S_t,k}(A)\|_F^2) \leq \|A - \pi_k(A)\|_F^2 + \frac{k}{cs}\|E\|_F^2.$$

Combining this inequality with the fact that $\|E\|_F^2 \leq \|A - \pi_{S_1 \cup \cdots \cup S_{t-1},k}(A)\|_F^2$ we get

$$\begin{aligned} &\mathsf{E}_{S_t}(\|A - \pi_{S_1 \cup \cdots \cup S_t,k}(A)\|_F^2) \\ &\quad \leq \|A - \pi_k(A)\|_F^2 + \frac{k}{cs}\|A - \pi_{S_1 \cup \cdots \cup S_{t-1},k}(A)\|_F^2. \end{aligned}$$

Taking the expectation over $S_1, \ldots, S_{t-1}$:

$$\begin{aligned} &\mathsf{E}_S(\|A - \pi_{S_1 \cup \cdots \cup S_t,k}(A)\|_F^2) \\ &\quad \leq \|A - \pi_k(A)\|_F^2 + \frac{k}{cs}\mathsf{E}_{S_1,\ldots,S_{t-1}}\left(\|A - \pi_{S_1 \cup \cdots \cup S_{t-1},k}(A)\|_F^2\right) \end{aligned}$$

and the result follows from the induction hypothesis for $t - 1$. □

This adaptive sampling scheme suggests the following algorithm that makes $2t$ passes through the data and computes and a rank-k approximation within additive error ϵ^t.

Iterative Fast SVD

Input: $A \in \boldsymbol{R}^{m \times n}$ with M non-zero entries, integers $k \leq m$, t, error $\epsilon > 0$.
Output: A set of k vectors in $\boldsymbol{R}^n$.

1. Let $S = \emptyset$, $s = k/\epsilon$.

2. Repeat t times:
   (a) Let $E = A - \pi_S(A)$.
   (b) Let T be a sample of s rows of A according to the distribution that assigns probability $\frac{\|E^{(i)}\|^2}{\|E\|_F^2}$ to row i.
   (c) Let $S = S \cup T$.
3. Let $h_1, \ldots, h_k$ be the top k right singular vectors of $\pi_S(A)$.

Theorem 7.4. Algorithm **Iterative Fast SVD** finds vectors $h_1 \ldots, h_k \in \boldsymbol{R}^n$ such that their span V satisfies

$$\mathsf{E}\left(\|A - \pi_V(A)\|_F^2\right) \leq \frac{1}{1-\epsilon}\|A - \pi_k(A)\|_F^2 + \epsilon^t \|A\|_F^2. \tag{7.12}$$

The running time is $O\big(M\frac{kt}{\epsilon} + (m+n)\frac{k^2t^2}{\epsilon^2}\big)$.

Proof. For the correctness, observe that $\pi_V(A)$ is a random variable with the same distribution as $\pi_{S,k}(A)$ as defined in Theorem 7.3. Also, $\|A - \pi_{S,k}(A)\|_F^2 - \|A - \pi_k(A)\|_F^2$ is a non-negative random variable and Theorem 7.3 gives a bound on its expectation:

$$\begin{aligned}&\mathsf{E}_S(\|A - \pi_{S,k}(A)\|_F^2 - \|A - \pi_k(A)\|_F^2)\\ &\quad\leq \frac{\epsilon}{1-\epsilon}\|A - \pi_k(A)\|_F^2 + \epsilon^t\|A\|_F^2.\end{aligned}$$

We will now bound the running time. We maintain a basis of the rows indexed by S. In each iteration, we extend this basis orthogonally with a new set of vectors Y, so that it spans the new sample T. The residual squared length of each row, $\|E^{(i)}\|^2$, as well as the total, $\|E\|_F^2$, is computed by subtracting the contribution of $\pi_T(A)$ from the values that they had during the previous iteration. In each iteration, the projection onto Y needed for computing this contribution takes time $O(Ms)$. In iteration i, the computation of the orthonormal basis Y takes time $O(ns^2 i)$ (Gram–Schmidt orthonormalization of s vectors in $\boldsymbol{R}^n$ against an orthonormal basis of size at most $s(i+1)$). Thus, the total time in iteration i is $O(Ms + ns^2 i)$; with t iterations, this is $O(Mst + ns^2t^2)$. At the end of Step 7.1 we

have $\pi_S(A)$ in terms of our basis (an $m \times st$ matrix). Finding the top k singular vectors in Step 7.1 takes time $O(ms^2t^2)$. Bringing them back to the original basis takes time $O(nkst)$. Thus, the total running time is $O(Mst + ns^2t^2 + ms^2t^2 + nkst)$ or, in other words, $O\big(M\frac{kt}{\epsilon} + (m+n)\frac{k^2t^2}{\epsilon^2}\big)$. □

7.2 Volume Sampling

Volume sampling is a generalization of length-squared sampling. We pick subsets of k rows instead picking rows one by one. The probability that we pick a subset S is proportional to the volume of the k-simplex $\Delta(S)$ spanned by these k rows along with the origin. This method will give us a factor $(k+1)$ approximation (in expectation) and a proof that any matrix has k rows whose span contains such an approximation. Moreover, this bound is tight, i.e., there exist matrices for which no k rows can give a better approximation.

Theorem 7.5. Let S be a random subset of k rows of a given matrix A chosen with probability

$$P_S = \frac{\mathrm{Vol}(\Delta(S))^2}{\sum_{T:|T|=k}\mathrm{Vol}(\Delta(T))^2}.$$

Then $\tilde{A}_k$, the projection of A to the span of S, satisfies

$$\mathsf{E}\left(\|A - \tilde{A}_k\|_F^2\right) \leq (k+1)\|A - A_k\|_F^2.$$

Proof. For every $S \subseteq [m]$, let Δ_S be the simplex formed by formed by $\{A^{(i)} | i \in S\}$ and the origin, and let H_S be the linear subspace spanned by these rows.

$$\begin{aligned}\sum_{S,|S|=k+1}\mathrm{Vol}_{k+1}(\Delta_S)^2 &= \frac{1}{k+1}\sum_{S,|S|=k}\sum_{j=1}^{m}\frac{1}{(k+1)^2}\mathrm{Vol}_k(\Delta_S)^2 d(A^{(j)}, H_S)^2 \\ &= \frac{1}{(k+1)^3}\sum_{S,|S|=k}\mathrm{Vol}_k(\Delta_S)^2\ \sum_{j=1}^{m} d(A^{(j)}, H_S)^2\end{aligned}$$

Let $\sigma_1, \ldots, \sigma_n$ be the singular values of A. Then, using Lemma 7.6 (proved next), we can rewrite this as follows:

$$\begin{aligned}
&\frac{1}{((k+1)!)^2} \sum_{1 \le t_1 < \ldots < t_{k+1} \le n} \sigma_{t_1}^2 \ldots \sigma_{t_{k+1}}^2 \\
&\quad = \frac{1}{(k+1)^3} \sum_{S, |S|=k} \mathrm{Vol}_k(\Delta_S)^2 \sum_{j=1}^{m} d(A^{(j)}, H_S)^2
\end{aligned}$$

which means that

$$\begin{aligned}
&\sum_{S, |S|=k} \mathrm{Vol}_k(\Delta_S)^2 \|A - \pi_{S,k}(A)\|_F^2 \\
&\quad = \frac{k+1}{(k!)^2} \sum_{1 \le t_1 < \ldots < t_{k+1} \le n} \sigma_{t_1}^2 \ldots \sigma_{t_{k+1}}^2 \\
&\quad \le \frac{k+1}{(k!)^2} \sum_{1 \le t_1 < \ldots < t_k \le n} \sigma_{t_1}^2 \ldots \sigma_{t_k}^2 \sum_{j=k+1}^{m} \sigma_j^2 \\
&\quad \le \left(\sum_{S, |S|=k} \mathrm{Vol}_k(\Delta_S)^2 \right) (k+1) \|A - A_k\|_F^2
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\frac{1}{\left(\sum_{S, |S|=k} \mathrm{Vol}_k(\Delta_S)^2 \right)} \sum_{S, |S|=k} \mathrm{Vol}_k(\Delta_S)^2 \|A - \pi_{S,k}(A)\|_F^2 \\
&\quad \le (k+1) \|A - A_k\|_F^2
\end{aligned}$$

And therefore there must exist a set S of k rows of A such that

$$\|A - \pi_{S,k}(A)\|_F^2 \le (k+1) \|A - A_k\|_F^2.$$

The coefficient of $\|A - \pi_{S,k}(A)\|_F^2$ on the LHS is precisely the probability with which S is chosen by volume sampling. Hence,

$$\mathsf{E}\left(\|A - \pi_{S,k}(A)\|_F^2 \right) \le (k+1) \|A - A_k\|_F^2. \qquad \square$$

Lemma 7.6.

$$\sum_{S,|S|=k} \mathrm{Vol}_k(\Delta_S)^2 = \frac{1}{(k!)^2} \sum_{1\le t_1<t_2<\ldots<t_k\le n} \sigma_{t_1}^2\sigma_{t_2}^2\ldots\sigma_{t_k}^2$$

where $\sigma_1, \sigma_2, \ldots, \sigma_n$ are the singular values of A.

Proof. Let A_S be the sub-matrix of A formed by the rows $\{A^{(i)} \mid i \in S\}$. Then we know that the volume of the k-simplex formed by these rows is given by

$$\mathrm{Vol}_k(\Delta_S) = \frac{1}{k!}\sqrt{det(A_S A_S^T)}$$

Therefore,

$$\begin{aligned}\sum_{S,|S|=k} \mathrm{Vol}_k(\Delta_S)^2 &= \frac{1}{(k!)^2} \sum_{S,|S|=k} det(A_S A_S^T) \\ &= \frac{1}{(k!)^2} \sum_{\substack{B \text{ : principal} \\ k\text{-minor of } AA^T}} det(B)\end{aligned}$$

Let $det(AA^T - \lambda I) = \lambda^m + c_{m-1}\lambda^{m-1} + \cdots + c_0$ be the characteristic polynomial of AA^T. From basic linear algebra we know that the roots of this polynomial are precisely the eigenvalues of AA^T, i.e., $\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2$ and 0 with multiplicity $(m-n)$. Moreover, the coefficient c_{m-k} can be expressed in terms of these roots as:

$$c_{m-k} = (-1)^{m-k} \sum_{1\le t_1<t_2<\cdots<t_k\le n} \sigma_{t_1}^2\sigma_{t_2}^2\cdots\sigma_{t_k}^2$$

But we also know that c_{m-k} is the coefficient of λ^{m-k} in $det(AA^T - \lambda I)$, which by Lemma 7.7 is

$$c_{m-k} = (-1)^{m-k} \sum_{\substack{B \text{ : principal} \\ k\text{-minor of} AA^T}} det(B)$$

Therefore,

$$\sum_{S,|S|=k} \mathrm{Vol}_k(\Delta_S)^2 = \frac{1}{(k!)^2} \sum_{1\le t_1<t_2<\cdots<t_k\le n} \sigma_{t_1}^2\sigma_{t_2}^2\cdots\sigma_{t_k}^2 \qquad \square$$

Lemma 7.7. Let the characteristic polynomial of $M \in \boldsymbol{R}^{m\times m}$ be $det(M - \lambda I_m) = \lambda^m + c_{m-1}\lambda^{m-1} + \cdots + c_0$. Then

$$c_{m-k} = (-1)^{m-k} \sum_{\substack{B, B \text{ principal} \\ k-\text{minor of } M}} det(B) \quad for\ 1 \leq k \leq m$$

Proof. We use the following notation. Let $M' = M - \lambda I$, and S_m be the set of permutation of $\{1, 2, \ldots, m\}$. The sign of a permutation $sgn(\tau)$, for $\tau \in Perm([m])$, is equal to 1 if it can be written as a product of an even number of transpositions and -1 otherwise. For a subset S of rows, we denote the submatrix of entries $(M_{i,j})_{i,j\in S}$ by M_S.

$$\begin{aligned} det(M - \lambda I_m) &= det(M') \\ &= \sum_{\tau\in Perm([m])} sgn(\tau) M'_{1,\tau(1)} M'_{2,\tau(2)} \cdots M'_{m,\tau(m)} \end{aligned}$$

The term $c_{m-k}\lambda^{m-k}$ comes by taking sum over τ which fix some set $S \subseteq [m]$ of size $(m-k)$, and the elements $\prod_{i\in S} M'_{i,i}$ contribute $(-1)^{m-k}\lambda^{m-k}$ and the coefficient comes from the constant term in $\sum_{\tau\in Perm([m]-S)} sgn(\tau) \prod_{i\notin S} M'_{i,\tau(i)}$. This, by induction hypothesis, is equal to $\sum_{S,|S|=m-k} det(M_{[m]-S})$. Hence

$$c_{m-k} = (-1)^{m-k} \sum_{S,|S|=m-k} det(M_{[m]-S}) = (-1)^{m-k} \sum_{\substack{B, B \text{ principal} \\ k-\text{minor of } M}} det(B)$$

□

Volume sampling leads to the following existence result for interpolative low-rank approximation.

Theorem 7.8. Any matrix A contains a set of $2k\log(k+1) + (4k/\epsilon)$ rows in whose span lies a rank-k matrix $\tilde{A}_k$ with the property that

$$\|A - \tilde{A}_k\|_F^2 \leq (1+\epsilon)\|A - A_k\|_F^2.$$

The proof follows from using Theorem 7.5 followed by multiple rounds of adaptive length-squared sampling.

Exercise 7.9. Prove Theorem 7.8.

The next exercise gives a fast procedure that approximates the volume sampling distribution.

Exercise 7.10. Let S be a subset of k rows of a given matrix A generated as follows: The first row is picked from $LS_{row(A)}$. The ith row is picked from $LS_{row(\hat{A}^i)}$ where $\hat{A}^i$ is the projection of A orthogonal to the span of the first $i-1$ rows chosen.

1. Show that
$$\mathsf{E}\left(\|A - \pi_S(A)\|_F^2\right) \leq (k+1)!\|A - A_k\|_F^2.$$
2. As in Exercise 7.9, use adaptive length-squared sampling to reduce the error to $(1+\epsilon)$. What is the overall time complexity and the total number of rows sampled?

7.2.1 A Lower Bound

The following proposition shows that Theorem 7.5 is tight.

Proposition 7.11. Given any $\epsilon > 0$, there exists a $(k+1) \times (k+1)$ matrix A such that for any subset S of k rows of A,
$$\|A - \pi_{S,k}(A)\|_F^2 \geq (1-\epsilon)\ (k+1)\ \|A - A_k\|_F^2$$

Proof. The tight example consists of a matrix with $k+1$ rows which are the vertices of a regular k-dimensional simplex lying on the affine hyperplane $\{X_{k+1} = \alpha\}$ in $\boldsymbol{R}^{k+1}$. Let $A^{(1)}, A^{(2)}, \ldots, A^{(k+1)}$ be the vertices with the point $p = (0, 0, \ldots, 0, \alpha)$ as their centroid. For α small enough, the best k-dimensional subspace for these points is given by $\{X_{k+1} = 0\}$ and
$$\|A - A_k\|_F^2 = (k+1)\alpha^2$$

Consider any subset of k points from these, say $S = \{A^{(1)}, A^{(2)}, \ldots, A^{(k)}\}$, and let H_S be the linear subspace spanning them. Then,

$$\|A - \pi_{S,k}(A)\|_F^2 = d(A^{(k+1)}, H_S)^2.$$

We claim that for any $\epsilon > 0$, α can be chosen small enough so that

$$d(A^{(k+1)}, H_S) \geq \sqrt{(1-\epsilon)}(k+1)\alpha.$$

Choose α small enough so that $d(p, H_S) \geq \sqrt{(1-\epsilon)}\alpha$. Now

$$\frac{d(A^{(k+1)}, H_S)}{d(p, H_S)} = \frac{d(A^{(k+1)}, \text{conv}(A^{(1)}, \ldots, A^{(k)}))}{d(p, \text{conv}(A^{(1)}, \ldots, A^{(k)}))} = k+1$$

since the points form a simplex and p is their centroid. The claim follows. Hence,

$$\begin{aligned} \|A - \pi_{S,k}(A)\|_F^2 &= d(A^{(k+1)}, H_S)^2 \\ &\geq (1-\epsilon)(k+1)^2\alpha^2 \\ &= (1-\epsilon)\ (k+1)\ \|A - A_k\|_F^2 \end{aligned}$$ □

Exercise 7.12. Extend the above lower bound to show that for $0 \leq \epsilon \leq 1/2$, there exist matrices for which one needs $\Omega(k/\epsilon)$ rows to span a rank-k matrix that is a $(1+\epsilon)$ approximation.

7.3 Isotropic Random Projection

In this section, we describe another randomized algorithm which also gives relative approximations to the optimal rank-k matrix with roughly the same time complexity. Moreover, it makes only *two* passes over the input data.

The idea behind the algorithm can be understood by going back to the matrix multiplication algorithm described in Section 6. There to multiply two matrices A, B, we picked random columns of A and rows of B and thus derived an estimate for AB from these samples. The error bound derived was additive and this is unavoidable. Suppose that we first project the rows of A randomly to a low-dimensional subspace, i.e., compute AR where R is random and $n \times k$, and similarly project

the columns of B, then we can use the estimate ARR^TB. For low-rank approximation, the idea extends naturally: first project the rows of A using a random matrix R, then project A to the span of the columns of AR (which is low dimensional), and finally find the best rank-k approximation of this projection.

Isotropic RP

Input: $A \in \boldsymbol{R}^{m \times n}$ with M non-zero entries, integers $k \leq m$, error $\epsilon > 0$.
Output: A rank-k matrix $\tilde{A}_k$.

1. Let $l = Ck/\epsilon$ and S be a random $l \times n$ matrix; compute $B = SA$.
2. Project A o the span of the rows of B to get $\tilde{A}$.
3. Output $\tilde{A}_k$, the best rank-k approximation of $\tilde{A}$.

Theorem 7.13. Let A be an $m \times n$ real matrix with M nonzeros. Let $0 < \epsilon < 1$ and S be an $r \times n$ random matrix with i.i.d. Bernoulli entries with mean zero and $r \geq Ck/\epsilon$ where C is a universal constant. Then with probability at least 3/4,

$$\|A - \pi_{SA,k}(A)\|_F \leq (1+\epsilon)\|A - A_k\|_F$$

and the singular vectors spanning $\pi_{SA,k}(A)$ can be computed in two passes over the data in $O(Mr + (m+n)r^2)$ time using $O((m+n)r^2)$ space.

Proof. (Outline) Consider the rank-k matrix $D = A_k V V^T$ where $SA = U\Sigma V^T$ is the SVD of SA. The rows of D lie in the span of the rows of SA. Hence,

$$\|A - \pi_{SA,k}A\|_F^2 \leq \|A - D\|_F^2 = \|A - A_k\|_F^2 + \|A_k - D\|_F^2.$$

We will now show that

$$\|A_k - D\|_F^2 \leq 2\epsilon\|A - A_k\|_F^2$$

which completes the proof.

To see this, we can view each row of $A - A_k$ as a linear regression problem, namely,

$$\min_x \|A^{(j)} - A_k x\|$$

for $j = 1, \ldots, n$ and let $x_1, \ldots, x_n$ be the solutions. The best approximation of $A^{(j)}$ from the row span of A_k is $A_k^{(j)}$. For a general linear regression problem,

$$\min_x \|Ax - b\|$$

the solution is $x = A^+ b$ where if $A = \hat{U}\hat{\Sigma}\hat{V}^T$ is the SVD of A, then $A^+ = \hat{V}\hat{\Sigma}^{-1}\hat{U}^T$ (see Exercise 7.14). Now consider the linear regressions

$$\min_x \|(SA)^{(j)} - (SA_k)x\|$$

for $j = 1, \ldots, n$. Let their solutions be $\tilde{x}_1, \ldots, \tilde{x}_n$. Then, there exist vectors $w_1, \ldots w_n$ orthogonal to the column span of U_k and $\beta_1, \ldots, \beta_n \in \boldsymbol{R}^k$ such that

$$\begin{aligned} w_j &= A^{(j)} - A_k^{(j)} \\ U\beta_j &= A_k \tilde{x}_j - A_k x_j \end{aligned}$$

From this (through a series of computations), we have, for $j = 1, \ldots, n$,

$$(U_k^T S^T S U_k)\beta_j = U_k^T S^T S w_j$$

Now we choose r large enough so that $\sigma^2(SU) \geq 1/\sqrt{2}$ with probability at least $7/8$ and hence,

$$\begin{aligned} \|A - D\|_F^2 &= \sum_{j=1}^n \beta_j^2 \\ &\leq 2\sum_{i=1}^n \|U_k^T S^T S w_j\|^2 \\ &\leq 2\epsilon \sum_{j=1}^n \|w_j\|^2 \\ &= 2\epsilon \sum_{j=1}^n \|A - A_k\|_F^2. \end{aligned}$$

Here the penultimate step we used the fact that random projection preserves inner products approximately, i.e., given that w_j is orthogonal to U_k,

$$|U_k^T S^T S w_j| \leq \epsilon^2 \|w_j\|^2. \qquad \square$$

Exercise 7.14. Let A be an $m \times n$ matrix with $m > n$ and $A = U\Sigma V^T$ be its SVD. Let $b \in \boldsymbol{R}^m$. Then the point x^* which minimizes $\|Ax - b\|$ is given by $x^* = V\Sigma^{-1}U^T b$.

7.4 Discussion

In this chapter we saw asymptotically tight bounds on the number of rows/columns whose span contains a near-optimal rank-k approximation of a given matrix. We also saw two different algorithms for obtaining such an approximation efficiently. Adaptive sampling was introduced in [29], volume sampling in [30] and isotropic RP in [61].

The existence of such sparse interpolative approximations has a nice application to clustering. Given a set of points in $\boldsymbol{R}^n$, and integers j, k, the projective clustering problem asks for a set of j k-dimensional subspaces such that the sum of squared distances of each point to its nearest subspace is minimized. Other objective functions, e.g., maximum distance or sum of distances has also been studied. The interpolative approximation suggests a simple enumerative algorithm: the optimal set of subspaces induce a partition of the point set; for each part, the subspace is given by the best rank-k approximation of the subset (the SVD subspace). From the theorems of this chapter, we know that a good approximation to the latter lies in the span of a small number (k/ϵ) of points. So, we simply enumerate over all subsets of points of this size, choosing j of them at a time. For each such choice, we have to consider all "distinct" k-dimensional subspaces in their span. This can be achieved by a discrete set of subspaces of exponential size, but only in k and ϵ. For each choice of j k-dimensional subspaces we compute the value of the objective function and output the minimum overall.

It is an open question to implement exact volume sampling efficiently, i.e., in time polynomial in both n and k. Another open question is to approximate a given matrix efficiently (nearly linear time or better) while incurring low error in the spectral norm.

8

Extensions of SVD

In this chapter, we discuss two extensions of SVD which provide substantial improvements or breakthroughs for some problems. The first is an extension of low-rank approximation from matrices to tensors (used in Section 5). Then we study an affine-invariant version of PCA, called *Isotropic PCA*. At first glance, this appears to be a contradiction in terms; however, there is a natural definition with applications (learning mixtures).

8.1 Tensor Decomposition via Sampling

We recall the basic set up. Corresponding to an r-dimensional tensor A, there is an r-linear form which for a set of r vectors, $x^{(1)}, x^{(2)}, \ldots, x^{(r-1)}, x^{(r)} \in \boldsymbol{R}^n$, is defined as

$$A(x^{(1)}, x^{(2)}, \ldots, x^{(r)}) = \sum_{i_1, i_2, \ldots, i_r} A_{i_1, i_2, \ldots, i_{r-1}, i_r} x^{(1)}_{i_1} x^{(2)}_{i_2}, \ldots x^{(r)}_{i_r}.$$

Recall the two norms of interest for tensors, the Frobenius norm and the 2-norm:

$$\|A\|_F = \left(\sum A^2_{i_1,i_2,\ldots,i_r}\right)^{\frac{1}{2}}$$
$$\|A\|_2 = \max_{x^{(1)},x^{(2)},\ldots,x^{(r)}} \frac{A(x^{(1)},x^{(2)},\ldots,x^{(r-1)},x^{(r)})}{|x^{(1)}||x^{(2)}|\ldots}.$$

We begin with the existence of a low-rank tensor decomposition.

Lemma 8.1. For any tensor A, and any $\epsilon > 0$, there exist $k \leq 1/\epsilon^2$ rank-1 tensors, $B_1, B_2, \ldots, B_k$ such that

$$\|A - (B_1 + B_2 + \ldots B_k)\|_2 \leq \epsilon \|A\|_F.$$

Proof. If $\|A\|_2 \leq \epsilon\|A\|_F$, then we are done. If not, there are vectors $x^{(1)}, x^{(2)}, \ldots, x^{(r)}$, all of length 1 such that

$$A(x^{(1)},x^{(2)},\ldots,x^{(r)}) \geq \epsilon\|A\|_F.$$

Now consider the r-dimensional array

$$B = A - (A(x^{(1)},x^{(2)},\ldots,x^{(r)}))x^{(1)} \otimes x^{(2)} \otimes \ldots x^{(r)}.$$

It is easy to see that

$$\|B\|_F^2 = \|A\|_F^2 - A(x^{(1)},x^{(2)},\ldots,x^{(r)})^2.$$

We can repeat on B and clearly this process will only go on for at most $1/\epsilon^2$ steps. □

Recall that for any $r-1$ vectors $x^{(1)}, x^{(2)}, \ldots, x^{(r-1)}$, the vector $A(x^{(1)}, x^{(2)}, \ldots, x^{(r-1)}, \cdot)$ has i-th component

$$\sum_{i_1,i_2,\ldots,i_{r-1}} A_{i_1,i_2,\ldots,i_{r-1},i} x^{(1)}_{i_1} x^{(2)}_{i_2}, \ldots x^{(r-1)}_{i_{r-1}}.$$

We now present an algorithm to solve the following problem: Given an r-dimensional tensor A, find unit vectors $x^{(1)}, x^{(2)}, \ldots, x^{(r)}$ maximizing $A(x^{(1)}, x^{(2)}, \ldots, x^{(r)})$ to within *additive error* $\epsilon\|A\|_F/2$.

Tensor decomposition

Set $\eta = \epsilon^2/100r\sqrt{n}$ and $s = 10^5 r^3/\epsilon^2$.

1. Pick s random $(r-1)$-tuples $(i_1, i_2, \ldots, i_{r-1})$ with probabilities proportional to the sum of squared entries on the line defined by it:

$$p(i_1, i_2, \ldots, i_{r-1}) = \frac{\sum_i A^2_{i_1,i_2,\ldots,i_{r-1},i}}{\|A\|_F^2}.$$

   Let I be the set of s $r-1$ tuples picked.

2. For each $i_1, i_2, \ldots, i_{r-1} \in I$, enumerate all possible values of $\hat{z}^{(1)}_{i_1}, \hat{z}^{(2)}_{i_2}, \ldots, \hat{z}^{(r-1)}_{i_{r-1}}$ whose coordinates are in the set

$$J = \{-1, -1+\eta, -1+2\eta, \ldots 0, \ldots, 1-\eta, 1\}^{s(r-1)}.$$

   (a) For each set of $\hat{z}^{(t)}$, for each $i \in V_r$, compute

$$y_i = \sum_{(i_1,\ldots,i_{r-1})\in I} A(i_1, \ldots, i_{r-1}, i)\hat{z}^{(1)}_{i_1} \ldots \hat{z}^{(r-1)}_{i_{r-1}}.$$

   and normalize the resulting vector y to be a unit vector.

   (b) Consider the $(r-1)$-dimensional array $A(y)$ defined by

$$(A(y))_{i_1,i_2,\ldots,i_{r-1}} = \sum_i A_{i_1,i_2,i_3\ldots i_{r-1},i} \quad y_i$$

   and apply the algorithm recursively to find the optimum

$$A(y)(x^{(1)}, x^{(2)}, \ldots, x^{(r-1)})$$

   with $|x^{(1)}| = \ldots |x^{(r-1)}| = 1$ to within additive error $\epsilon\|A(y)\|_F/2$.

3. Output the set of vectors that gives the maximum among all the candidates.

To see the idea behind the algorithm, let $z^{(1)}, z^{(2)}, \ldots, z^{(r)}$ be unit vectors that maximize $A(x^{(1)}, x^{(2)}, \ldots, x^{(r)})$. Since

$$A(z^{(1)}, \ldots, z^{(r-1)}, z^{(r)}) = z^{(r)} \cdot A(z^{(1)}, \ldots, z^{(r-1)}, \cdot),$$

we have

$$z^{(r)} = \frac{A(z^{(1)}, z^{(2)}, \ldots, z^{(r-1)}, \cdot)}{|A(z^{(1)}, z^{(2)}, \ldots, z^{(r-1)}, \cdot)|}.$$

Thus, $z^{(r)}$ is a function of $z^{(1)}, z^{(2)}, \ldots, z^{(r-1)}$. Therefore, we can estimate the components of $z^{(r)}$ given random terms in the sum $A(z^{(1)}, \ldots, z^{(r-1)}, \cdot)$. We will need only $s = O(r^3/\epsilon^2)$ terms for a good estimate. Also, we do not need to know the $z^{(1)}, z^{(2)}, \ldots, z^{(r-1)}$ completely; only $s(r-1)$ of coordinates in total will suffice. We enumerate all possibilities for the values of these coordinates. For each candidate $z^{(r)}$, we can reduce the problem to maximizing an $(r-1)$-dimensional tensor and we solve this recursively. We then take the best candidate set of vectors.

We proceed to analyze the algorithm and prove the following theorem.

Theorem 8.2. For any tensor A, and any $\epsilon > 0$, we can find k rank-1 tensors $B_1, B_2, \ldots, B_k$, where $k \leq 4/\epsilon^2$, in time $(n/\epsilon)^{O(1/\epsilon^4)}$ such that with probability at least 3/4 we have

$$\|A - (B_1 + B_2 + \ldots B_k)\|_2 \leq \epsilon \|A\|_F.$$

For $r = 2$, the running time can be improved to a fixed polynomial in n and exponential only in $(1/\epsilon)$. We begin by bounding the error introduced by the discretization.

Lemma 8.3. Let $z^{(1)}, z^{(2)}, \ldots, z^{(r-1)}$ be the optimal unit vectors. Suppose $w^{(1)}, w^{(2)}, \ldots, w^{(r-1)}$ are obtained from the zs by rounding each coordinate down to the nearest integer multiple of η, with $0 \leq \eta < 1$. Then,

$$\left| A(z^{(1)}, \ldots, z^{(r-1)}, \cdot) - A(w^{(1)}, \ldots, w^{(r-1)}, \cdot) \right| \leq \eta r \sqrt{n} \|A\|_F.$$

Proof. We can write

$$\begin{aligned}
&\left|A(z^{(1)}, z^{(2)}, \ldots, z^{(r-1)}, \cdot) - A(w^{(1)}, w^{(2)}, \ldots, w^{(r-1)}, \cdot)\right| \\
&\quad \leq \left|A(z^{(1)}, z^{(2)}, \ldots, z^{(r-1)}, \cdot) - A(w^{(1)}, z^{(2)}, \ldots, z^{(r-1)}, \cdot)\right| \\
&\qquad + \left|A(w^{(1)}, z^{(2)}, \ldots, z^{(r-1)}, \cdot) - A(w^{(1)}, w^{(2)}, z^{(3)}, \ldots, z^{(r-1)}, \cdot)\right| \ldots
\end{aligned}$$

A typical term above is

$$\begin{aligned}
&\left|A\big(w^{(1)}, \ldots, w^{(t)}, z^{(t+1)}, \ldots, z^{(r-1)}, \cdot\big)\right. \\
&\qquad \left. - A\big(w^{(1)}, \ldots, w^{(t)}, w^{(t+1)}, z^{(t+2)}, \ldots, z^{(r-1)}, \cdot\big)\right| \\
&\qquad\quad \leq \left|C(z^{(t+1)} - w^{(t+1)})\right| \\
&\qquad\quad \leq \|C\|_2 |z^{(t+1)} - w^{(t+1)}| \\
&\qquad\quad \leq \|C\|_F \eta\sqrt{n} \leq \|A\|_F \eta\sqrt{n}.
\end{aligned}$$

Here, C is the matrix defined as the matrix whose ij'th entry is

$$\sum_{j_1, \ldots j_t, j_{t+2} \ldots j_{r-1}} A_{j_1, \ldots, j_t, i, j_{t+2}, \ldots, j_{r-1}, j} w^{(1)}_{j_1} \ldots w^{(t)}_{j_t} z^{(t+2)}_{j_{t+2}} \ldots z^{(r-1)}_{j_{r-1}}$$

The claim follows. □

We analyze the error incurred by sampling in the next two lemmas.

Lemma 8.4. For an $(r-1)$-tuple $(i_1, i_2, \ldots, i_{r-1}) \in I$, define the random variables variables X_i for $i = 1, \ldots, n$ by

$$X_i = \frac{A_{i_1, i_2, \ldots, i_{r-1}, i} w^{(1)}_{i_1} w^{(2)}_{i_2} \ldots w^{(r-1)}_{i_{r-1}}}{p(i_1, i_2, \ldots, i_{r-1})}.$$

Then,

$$\mathsf{E}\,(X_i) = A(w^{(1)}, w^{(2)} \ldots w^{(r-1)}, \cdot)_i.$$

and

$$\mathsf{Var}\,(X_i) \leq \|A\|_F^2.$$

Proof. The expectation is immediate, while the variance can be estimated as follows:

$$\begin{aligned}\sum_i \mathsf{Var}\,(X_i) &\le \sum_i \sum_{i_1,i_2,\ldots,} \frac{A^2_{i_1,i_2,\ldots,i_{r-1},i}\left(w^{(1)}_{i_1}\ldots w^{(r-1)}_{i_{r-1}}\right)^2}{p(i_1,i_2,\ldots,)} \\ &\le \sum_{i_1,i_2,\ldots,} \frac{\left(z^{(1)}_{i_1}\ldots z^{(r-1)}_{i_{r-1}}\right)^2}{p(i_1,i_2,\ldots,)} \sum_i A^2_{i_1,i_2,\ldots,i_{r-1},i} \\ &\le \|A\|_F^2.\end{aligned}$$

□

Lemma 8.5. Define

$$\zeta = A(z^{(1)}, z^{(2)}, \ldots, z^{(r-1)}, \cdot).$$

In the list of candidate vectors enumerated by the algorithm will be a vector y such that

$$\|A\left(\frac{y}{|y|}\right) - A\left(\frac{\zeta}{|\zeta|}\right)\|_F \le \frac{\epsilon}{10r}\|A\|_F.$$

Proof. Consider the vector y computed by the algorithm when all $\hat{z}^{(t)}$ are set to $w^{(t)}$, the rounded optimal vectors. This will clearly happen sometime during the enumeration. This y_i is just the sum of s i.i.d. copies of X_i, one for each element of I. Thus, we have that

$$E(y) = sA(w^{(1)}, w^{(2)}\ldots w^{(r-1)}, \cdot)$$

and

$$\mathsf{Var}\,(y) = E(|y - E(y)|^2) \le s\|A\|_F^2.$$

From the above, it follows that with probability at least $1 - (1/10r)$, we have

$$|\Delta| \le 10r\sqrt{s}\|A\|_F.$$

Using this,

$$\begin{aligned}\left|\frac{y}{|y|}-\frac{\zeta}{|\zeta|}\right| &= \frac{|(y|\zeta|-\zeta|y|)|}{|y||\zeta|}\\ &= \frac{1}{|y||\zeta|}\,|(\Delta+s\zeta)|\zeta|-\zeta(|y|-s|\zeta|+s|\zeta|)|\\ &\le \frac{2|\Delta|}{(s|y|)} \le \frac{\epsilon}{50r^2},\end{aligned}$$

assuming $|y| \ge \epsilon\|A\|_F/100r$. If this assumption does not hold, we know that the $|\zeta| \le \epsilon\|A\|_F/20r$ and in this case, the all-zero tensor is a good approximation to the optimum. From this, it follows that

$$\|A\left(\frac{y}{|y|}\right) - A\left(\frac{\zeta}{|\zeta|}\right)\|_F \le \frac{\epsilon}{10r}\|A\|_F. \qquad \square$$

Thus, for any $r-1$ unit length vectors $a^{(1)}, a^{(2)}, \ldots, a^{(r-1)}$, we have

$$\left|A\left(a^{(1)},\ldots,a^{(r-1)},\frac{y}{|y|}\right) - A\left(a^{(1)},\ldots,a^{(r-1)},\frac{\zeta}{|\zeta|}\right)\right| \le \frac{\epsilon}{10r}\|A\|_F.$$

In words, the optimal set of vectors for $A(y/|y|)$ is nearly optimal for $A(\zeta/|\zeta|)$. Since $z^{(r)} = \zeta/|\zeta|$, the optimal vectors for the latter problem are $z^{(1)}, \ldots, z^{(r-1)}$. Applying this argument at every phase of the algorithm, we get a bound on the total error of $\epsilon\|A\|_F/10$.

The running time of algorithm is dominated by the number of candidates we enumerate, and is at most

$$\text{poly}(n)\left(\frac{1}{\eta}\right)^{s^2 r} = \left(\frac{n}{\epsilon}\right)^{O(1/\epsilon^4)}.$$

This completes the proof of Theorem 8.2.

8.2 Isotropic PCA

In this section we discuss an extension of Principal Component Analysis (PCA) that is able to go beyond standard PCA in identifying

"important" directions. Suppose the covariance matrix of the input (distribution or point set in $\boldsymbol{R}^n$) is a multiple of the identity. Then, PCA reveals no information — the second moment along any direction is the same. The extension, called *isotropic PCA*, can reveal interesting information in such settings. In Section 2, we used this technique to give an affine-invariant clustering algorithm for points in $\boldsymbol{R}^n$. When applied to the problem of unraveling mixtures of arbitrary Gaussians from unlabeled samples, the algorithm yields strong guarantees.

To illustrate the technique, consider the uniform distribution on the set $X = \{(x,y) \in \mathbb{R}^2 : x \in \{-1,1\}, y \in [-\sqrt{3},\sqrt{3}]\}$, which is isotropic. Suppose this distribution is rotated in an unknown way and that we would like to recover the original x and y axes. For each point in a sample, we may project it to the unit circle and compute the covariance matrix of the resulting point set. The x direction will correspond to the greater eigenvector, the y direction to the other. Instead of projection onto the unit circle, this process may also be thought of as importance weighting, a technique which allows one to simulate one distribution with another. In this case, we are simulating a distribution over the set X, where the density function is proportional to $(1+y^2)^{-1}$, so that points near $(1,0)$ or $(-1,0)$ are more probable.

More generally, isotropic PCA first puts a given distribution in isotropic position, then reweights points using a spherically symmetric distribution and performs PCA on this reweighted distribution. The core of PCA is finding a direction that maximizes the second moment. When a distribution is isotropic, the second moment of a random point X is the same for any direction v, i.e., $\mathsf{E}\left((v^T X)^2\right)$ is constant. In this situation, one could look for directions which maximize higher moments, e.g., the fourth moment. However, finding such directions seems to be hard. Roughly speaking, isotropic PCA finds directions which maximize a certain weighted averages of higher moments.

In the description below, the input to the algorithm is an $m \times n$ matrix (rows are points in $\boldsymbol{R}^n$).

Isotropic PCA

1. Apply an isotropic transformation to the input data, so that the mean of the resulting data is zero and its covariance matrix is the identity.
2. Weight each point using the density of a spherically symmetric weight function centered at zero, e.g., a spherical Gaussian.
3. Perform PCA on the weighted data.

In the application to Gaussian mixtures, the reweighting density is indeed a spherical Gaussian.

8.3 Discussion

Tensors are natural generalizations of matrices and seem to appear in many data sets, e.g., network traffic (sender, receiver, time), or the Web (document, term, hyperlink). However, many algorithmic problems that can be solved efficiently for matrices appear to be harder or intractable. Even finding the vector that maximizes the spectral norm of a tensor is NP-hard. Thus, it seems important to understand what properties of tensors or classes of tensors are algorithmically useful. The sampling-based tensor approximation presented here is from [25].

Isotropic PCA was introduced in Brubaker and Vempala [15] and applied to learning mixtures. It would be interesting to see if other problems could be tackled using this tool. In particular, the directions identified by the procedure might have significance in convex geometry and functional analysis.

References

[1] D. Achlioptas and F. McSherry, "On Spectral Learning of Mixtures of Distributions," in *Proceedings of COLT*, 2005.

[2] D. Achlioptas and F. McSherry, "Fast computation of low-rank matrix approximations," *Journal of the ACM*, vol. 54, no. 2, 2007.

[3] D. Aloise, A. Deshpande, P. Hansen, and P. Popat, "NP-hardness of Euclidean sum-of-squares clustering," *Machine Learning*, vol. 75, no. 2, pp. 245–248, 2009.

[4] N. Alon, W. DeLaVega, R. Kannan, and M. Karpinski, "Random sub-problems of Max-SNP problems," *Proceedings of the 34th Annual ACM Symposium on Theory on Computing*, pp. 668–677, 2002.

[5] N. Alon, M. Krivelevich, and B. Sudakov, "Finding a large hidden clique in a random graph," *Random Structures and Algorithms*, vol. 13, pp. 457–466, 1998.

[6] S. Arora and R. Kannan, "Learning mixtures of arbitrary Gaussians," *Annals of Applied Probability*, vol. 15, no. 1A, pp. 69–92, 2005.

[7] S. Arora, D. Karger, and M. Karpinski, "Polynomial time approximation schemes for dense instances of NP-hard problems," *Proceedings of the 27th Annual ACM Symposium on Theory of Computing*, pp. 284–293, 1995.

[8] S. Arora, S. Rao, and U. Vazirani, "Expander flows, geometric embeddings and graph partitioning," in *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 222–231, 2004.

[9] D. Arthur and S. Vassilvitskii, "k-means++: The Advantages of Careful Seeding," in *Proceedings of SODA*, 2007.

[10] Y. Azar, A. Fiat, A. Karlin, and F. McSherry, "Spectral analysis of data," in *Proceedings of STOC*, pp. 619–626, 2001.

[11] R. Bhatia, "Matrix factorizations and their perturbations," *Linear Algebra and its applications*, vol. 197, 198, pp. 245–276, 1994.
[12] R. Bhatia, *Matrix Analysis*. Springer, 1997.
[13] R. Boppana, "Eigenvalues and graph bisection: An average-case analysis," in *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pp. 280–285, 1987.
[14] S. C. Brubaker, "Robust PCA and Clustering on Noisy Mixtures," in *Proceedings of SODA*, 2009.
[15] S. C. Brubaker and S. Vempala, "Isotropic PCA and affine-invariant clustering," in *Building Bridges Between Mathematics and Computer Science*, 19, (M. Grötschel and G. Katona, eds.), Bolyao Society Mathematical Studies, 2008.
[16] M. Charikar, S. Guha, Éva Tardos, and D. B. Shmoys, "A constant-factor approximation algorithm for the k-median problem," in *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pp. 1–10, 1999.
[17] K. Chaudhuri and S. Rao, "Beyond Gaussians: Spectral Methods for Learning Mixtures of Heavy-Tailed Distributions," in *Proceedings of COLT*, 2008.
[18] K. Chaudhuri and S. Rao, "Learning mixtures of product distributions using correlations and independence," in *Proceedings of COLT*, 2008.
[19] D. Cheng, R. Kannan, S. Vempala, and G. Wang, "A divide-and-merge methodology for clustering," *ACM Transactions on Database Systems*, vol. 31, no. 4, pp. 1499–1525, 2006.
[20] A. Dasgupta, J. Hopcroft, R. Kannan, and P. Mitra, "Spectral clustering with limited independence," in *Proceedings of SODA*, pp. 1036–1045, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics, 2007.
[21] A. Dasgupta, J. Hopcroft, J. Kleinberg, and M. Sandler, "On learning mixtures of heavy-tailed distributions," in *Proceedings of FOCS*, 2005.
[22] S. DasGupta, "Learning mixtures of Gaussians," in *Proceedings of FOCS*, 1999.
[23] S. DasGupta and L. Schulman, "A two-round variant of EM for Gaussian mixtures," in *Proceedings of UAI*, 2000.
[24] W. F. de-la Vega, "MAX-CUT has a randomized approximation scheme in dense graphs," *Random Structures and Algorithms*, vol. 8, pp. 187–199, 1996.
[25] W. F. de la Vega, M. Karpinski, R. Kannan, and S. Vempala, "Tensor decomposition and approximation schemes for constraint satisfaction problems," in *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pp. 747–754, 2005.
[26] W. F. de la Vega, M. Karpinski, and C. Kenyon, "Approximation schemes for metric bisection and partitioning," in *Proceedings of 15th ACM-SIAM SODA*, pp. 499–508, 2004.
[27] W. F. de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani, "Approximation schemes for clustering problems," in *Proceedings of 35th ACM STOC*, pp. 50–58, 2003.
[28] W. F. de la Vega and C. Kenyon, "A randomized approximation scheme for metric MAX-CUT," *Journal of Computer and System Sciences*, vol. 63, pp. 531–541, 2001.

[29] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang, "Matrix approximation and projective clustering via volume sampling," *Theory of Computing*, vol. 2, no. 1, pp. 225–247, 2006.

[30] A. Deshpande and S. Vempala, "Adaptive sampling and fast low-rank matrix approximation," in *APPROX-RANDOM*, pp. 292–303, 2006.

[31] P. Drineas and R. Kannan, "Fast Monte-Carlo algorithms for approximate matrix multiplication," in *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pp. 452–459, 2001.

[32] P. Drineas and R. Kannan, "Pass efficient algorithms for approximating large matrices," in *SODA '03: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 223–232, 2003.

[33] P. Drineas, R. Kannan, A. Frieze, S. Vempala, and V. Vinay, "Clustering large graphs via the singular value decomposition," *Machine Learning*, vol. 56, pp. 9–33, 2004.

[34] P. Drineas, R. Kannan, and M. Mahoney, "Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix," *SIAM Journal on Computing*, vol. 36, pp. 132–157, 2006.

[35] P. Drineas, R. Kannan, and M. Mahoney, "Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix," *SIAM Journal on Computing*, vol. 36, pp. 158–183, 2006.

[36] P. Drineas, R. Kannan, and M. Mahoney, "Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix," *SIAM Journal on Computing*, vol. 36, pp. 184–206, 2006.

[37] P. Drineas, I. Kerenidis, and P. Raghavan, "Competitive recommendation systems," *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pp. 82–90, 2002.

[38] R. O. Duda, P. Hart, and D. Stork, *Pattern Classification*. John Wiley & Sons, 2001.

[39] J. Feldman, R. A. Servedio, and R. O'Donnell, "PAC learning axis-aligned mixtures of Gaussians with no separation assumption," in *Proceedings of COLT*, pp. 20–34, 2006.

[40] A. Frieze and R. Kannan, "The regularity lemma and approximation schemes for dense problems," *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computing*, pp. 12–20, 1996.

[41] A. Frieze and R. Kannan, "MAX-CUT has a randomized approximation scheme in dense graphs," *Quick Approximation to matrices and applications*, vol. 19, no. 2, pp. 175–200, 1999.

[42] A. Frieze, R. Kannan, and S. Vempala, "Fast Monte-Carlo algorithms for finding low-rank approximations," in *Proceedings of FOCS*, pp. 370–378, 1998.

[43] A. Frieze, R. Kannan, and S. Vempala, "Fast Monte-Carlo algorithms for finding low-rank approximations," *Journal of the ACM*, vol. 51, no. 6, pp. 1025–1041, 2004.

[44] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[45] Z. Füredi and J. Komlós, "The eigenvalues of random symmetric matrices," *Combinatorica*, vol. 1, no. 3, pp. 233–241, 1981.

[46] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
[47] O. Goldreich, S. Goldwasser, and D. Ron, "Property testing and its connection to learning and approximation," *Journal of the ACM*, vol. 5, no. 4, pp. 653–750, 1998.
[48] G. H. Golub and C. F. van Loan, *Matrix Computations*. Johns Hopkins University Press, 3rd ed., 1996.
[49] S. Har-Peled and K. R. Varadarajan, "Projective clustering in high dimensions using core-sets," in *Symposium on Computational Geometry*, pp. 312–318, 2002.
[50] P. Indyk, "A sublinear time approximation scheme for clustering in metric spaces," in *Proceedings of 40th IEEE FOCS*, pp. 154–159, 1999.
[51] R. Kannan, H. Salmasian, and S. Vempala, "The spectral method for general mixture models," *SIAM Journal on Computing*, vol. 38, no. 3, pp. 1141–1156, 2008.
[52] R. Kannan, S. Vempala, and A. Vetta, "On clusterings: Good, bad and spectral," *Journal of ACM*, vol. 51, no. 3, pp. 497–515, 2004.
[53] J. A. Kelner, "Spectral partitioning, eigenvalue bounds, and circle packings for graphs of bounded genus," *SIAM Journal on Computing*, vol. 35, no. 4, pp. 882–902, 2006.
[54] F. T. Leighton and S. Rao, "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms," *Journal of the ACM*, vol. 46, no. 6, pp. 787–832, 1999.
[55] L. Lovász and S. Vempala, "The geometry of logconcave functions and sampling algorithms," *Random Structures and Algorithms*, vol. 30, no. 3, pp. 307–358, 2007.
[56] F. Lust-Piquard, "Inégalites de Khinchin dans $C_p(1 < p < \infty)$," *Comptes Rendus de l'Académie des sciences, Paris*, vol. 303, pp. 289–292, 1986.
[57] F. McSherry, "Spectral partitioning of random graphs," in *FOCS*, pp. 529–537, 2001.
[58] N. Megiddo and A. Tamir, "On the complexity of locating facilities in the plane," *Operations Research Letters*, vol. I, pp. 194–197, 1982.
[59] C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala, "Latent semantic indexing: A probabilistic analysis," in *Proceedings of PODS*, 1998.
[60] M. Rudelson, "Random vectors in the isotropic position," *Journal of Functional Analysis*, vol. 164, pp. 60–72, 1999.
[61] T. Sarlós, "Improved approximation algorithms for large matrices via random projections," in *FOCS*, pp. 143–152, 2006.
[62] A. Sinclair and M. Jerrum, "Approximate counting, uniform generation and rapidly mixing Markov chains," *Information and Computation*, vol. 82, pp. 93–133, 1989.
[63] D. A. Spielman and S.-H. Teng, "Spectral partitioning works: Planar graphs and finite element meshes," *Linear Algebra and its Applications*, vol. 421, no. 2–3, pp. 284–305, 2007.
[64] G. Strang, *Linear Algebra and Its Applications*. Brooks Cole, 1988.

[65] S. Vempala and G. Wang, "A spectral algorithm for learning mixtures of distributions," *Journal of Computer and System Sciences*, vol. 68, no. 4, pp. 841–860, 2004.

[66] V. H. Vu, "Spectral norm of random matrices," in *Proceedings of STOC*, pp. 423–430, 2005.

[67] J. Wilkinson, *The algebraic eigenvalue problem (paperback ed.)*. Oxford: Clarendon Press, 1988.