Spectral Algorithms for Data Analysis (draft)

Ravindran Kannan and Santosh S. Vempala

September 12, 2021

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

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

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

Contents

Part I

Applications

Chapter 1

The Best-Fit Subspace

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

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

1.1 Singular Value Decomposition

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

$$Av = \lambda v.$$

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

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

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

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

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

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

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

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

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

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

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

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

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

Proposition 1.1.

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

Proof. We simply use the identity:

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

□

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

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

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

Proof. For any singular vector v,

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

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

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

and thus

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

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

Now we can also write

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

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

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

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

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

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

□

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

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

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

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

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

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

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

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

Such a decomposition where,

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

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

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

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

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

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

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

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

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

We can verify that the decomposition

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

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

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

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

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

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

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

Proof. We have

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

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

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

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

1.2 Algorithms for computing the SVD

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

Assume that A is symmetric.

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

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

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

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

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

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

1.3 The k-means clustering problem

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Algorithm: Voronoi-k-means

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

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

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

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

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

Algorithm for the k-means clustering problem

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

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

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

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

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

Combining this with (1.3) we get

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

proving that we do indeed get a 2-approximation.

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

1.4 Discussion

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

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

Chapter 2

Unraveling Mixtures Models

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

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

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

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

$$F = w_1 F_1 + w_2 F_2 + \cdots + w_k F_k$$

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

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

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

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

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

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

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

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

2.1 The challenge of high dimensionality

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

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

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

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

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

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

2.2 Classifying separable mixtures

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

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

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

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

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

where

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

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

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

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

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

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

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

By Vajda's lower bound, we have

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

Then either

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

or

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

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

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

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

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

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

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

We consider separation of the form

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

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

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

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

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

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

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

for $m > 2$. Thus the inequality

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

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

respectively, with $i \neq j$,

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

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 random projection from $\boldsymbol{R}^n \to \boldsymbol{R}^\ell$ so that the image of a point u is u'. Then it can be shown that

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

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

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

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

2.2.1 Spectral projection

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

Algorithm: Classify-Mixture

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

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

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

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

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

Lemma 2.4. *The best-fit 1-dimensional subspace for a spherical Gaussian with mean μ is given by the vector passing through μ.*

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

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

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

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

Lemma 2.5. *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 2.6. *The k-dim SVD subspace for a mixture of k Gaussians F contains the span of $\{\mu_1, \mu_2, ..., \mu_k\}$.*

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

Theorem 2.7. *Given m samples drawn from a mixture of k Gaussians with pairwise mean separation*

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

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

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

1. What distributions does Theorem 2.6 extend to?
2. What about more general distributions?
3. What is the sample complexity?

2.2.2 Weakly isotropic mixtures

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

Definition 2.8. *Random variable $X \in \mathbb{R}^n$ has a* weakly isotropic *distribution with mean μ and variance σ^2 if*

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

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

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

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

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

The statement of Exercise 2.2 does not hold for arbitrary distributions, even for $k = 1$. Consider a non-spherical Gaussian random vector $X \in \mathbb{R}^2$, whose mean is $(0, 1)$ and whose variance along the x-axis is much larger than that

along the y-axis. Clearly the optimal 1-dimensional subspace for X (that maximizes the squared projection in expectation) is not the one passes through its mean μ; it is orthogonal to the mean. SVD applied after centering the mixture at the origin works for one Gaussian but breaks down for $k > 1$, even with (nonspherical) Gaussian components.

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

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

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

It is known that for any isotropic convex body, a random point X satisfies the following tail inequality (Lemma 2.10 later in this chapter):

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

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

2.2.3 Mixtures of general distributions

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

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

Theorem 2.9. *Let F be a mixture of arbitrary distributions $F_1, \ldots, F_k$. Let w_i be the mixing weight of F_i, μ_i be its mean and $\sigma^2_{i,W}$ be the maximum variance of F_i along directions in W, the k-dimensional SVD-subspace of F. Then*

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

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

Theorem 2.9 says that for a mixture of general distributions, the means do not move too much after projection to the SVD subspace. Note that the theorem does not solve the case of parallel pancakes, as it requires that the pancakes be separated by a factor proportional to their "radius" rather than their "thickness".

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

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

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

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

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

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

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

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

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

2.2.4 Spectral projection with samples

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

This is useful for two reasons:

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

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

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

or equivalently,

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

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

Lemma 2.10. *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,*

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

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.

2.3 Learning mixtures of spherical distributions

So far our efforts have been to partition the observed sample points. The other interesting problem proposed in the introduction of this chapter was to identify the values μ_i, σ_i and w_i. In this section, we will see that this is possible in polynomial time provided the means μ_i are linearly independent. We let Y denote a sample from the mixture F. Thus,

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

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

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

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

Therefore,

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

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

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

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

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

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

Therefore we can compute

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

We do not know the μ_i's, so finding a v orthogonal to them is not straightforward. However, if we compute the SVD of the $m \times n$ matrix containing our m samples, the top k singular vectors would be the best fit k-dimensional subspace (see theorem 1.4), and assuming the means are linearly independent, this is exactly $\text{span}\{\mu_1 \ldots \mu_k\}$.

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

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

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

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

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

Then the third moment for Y can be expressed as

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

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

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

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

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

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

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

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

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

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

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

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

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

Algorithm: Learning the parameters

1. Compute $M = E_S(Y \otimes Y)$ and its top k eigenvectors $v_1, \ldots v_k$. Let $\bar{\sigma} = \sigma_{k+1}(M)$.

2. Project the data to the span of $v_1, \ldots, v_k$. Decompose $(M - \bar{\sigma} I) = WW^T$, using the SVD, and compute $\tilde{S} = W^{-1}S$.

3. Find a vector $\bar{v}$ orthogonal to span$\{W^{-1}v_1, \ldots W^{-1}v_k\}$ and compute
$$\bar{u} = E_{\tilde{S}}(Y((Y - \mu_Y)\bar{v})^2)$$
and
$$T = E_{\tilde{S}}(Y \otimes Y \otimes Y) - \sum_j (\bar{u} \otimes e_j \otimes e_j + e_j \otimes \bar{u} \otimes e_j + e_j \otimes e_j \otimes \bar{u}).$$

4. Iteratively apply the tensor power method on T. That is repeteadly apply
$$x := \frac{T(., x, x)}{\|T(., x, x)\|}$$
until convergence. Then set $\tilde{\mu}_1 = T(x, x, x)x$ and $w_1 = 1/|\tilde{\mu}_1|^2$ and repeat with
$$T := T - w_i \tilde{\mu}_i \otimes \tilde{\mu}_i \otimes \tilde{\mu}_i$$
to recover $\tilde{\mu}_2 \ldots \tilde{\mu}_k$. To recover the variances, compute $\sigma_i^2 = \bar{u} \cdot \tilde{\mu}_i / w_i^{3/2}$.

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

Theorem 2.12. *Given M and T as above, if all the means $\mu_1 \ldots \mu_k$ are linearly independent, we can estimate all parameters of each distribution in polynomial time.*

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

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

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

Now for the third-order tensor

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

we have,

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

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

2.4 An affine-invariant algorithm

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

The first step of the algorithm is to place the mixture in isotropic position via an affine transformation. This has the effect of making the $(k-1)$-dimensional Fisher subspace, i.e., the one that minimizes the Fisher discriminant (the fraction of the variance of the mixture taken up the intra-component term; see Section 2.4.2 for a formal definition), the same as the subspace spanned by the means of the components (they only coincide in general in isotropic position), for *any* mixture. The rest of the algorithm identifies directions close to this subspace and uses them to cluster, without access to labels. Intuitively this is hard since after isotropy, standard PCA/SVD reveals no additional information. Before presenting the ideas and guarantees in more detail, we describe relevant related work.

As before, we assume we are given a lower bound w on the minimum mixing weight and k, the number of components. With high probability, Algorithm

UNRAVEL returns a hyperplane so that each halfspace encloses almost all of the probability mass of a single component and almost none of the other component.

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

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

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

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

2.4.1 Parallel Pancakes

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

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

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

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

2.4.2 Analysis

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

Theorem 2.13. *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 half-spaces 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 2.13 is affine-invariant and much weaker than conditions of the form $\|\mu_1 - \mu_2\| \gtrsim \max\{\sigma_{1,\max}, \sigma_{2,\max}\}$ that came up earlier in the chapter. We note that the separating direction need not be the intermean direction.

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

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

Mathematically, this is expressed as

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

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

Theorem 2.14. *There is an absolute constant C for which the following holds. Suppose that $\mathcal{F}$ is a mixture of two Gaussians such that there exists a direction p for which*

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

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

In words, the algorithm successfully unravels arbitrary Gaussians provided there exists a line along which the expected squared distance of a point to its component mean is smaller than the expected squared distance to the overall mean by roughly a $1/w^3$ factor. There is no dependence on the largest variances of the individual components, and the dependence on the ambient dimension is logarithmic. Thus the addition of extra dimensions, even with large variance, has little impact on the success of the algorithm. The algorithm and its analysis in terms of the Fisher discriminant have been generalized to $k > 2$ [BV08].

2.5 Discussion

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

A related line of work considers learning symmetric product distributions, where the coordinates are independent. Feldman et al [FSO06] have shown that

mixtures of axis-aligned Gaussians can be approximated without any separation assumption at all in time exponential in k. Chaudhuri and Rao [CR08a] have given a polynomial-time algorithm for clustering mixtures of product distributions (axis-aligned Gaussians) under mild separation conditions. A. Dasgupta et al [DHKS05] and later Chaudhuri and Rao [CR08b] gave algorithms for clustering mixtures of heavy-tailed distributions.

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

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

A more general question is "agnostic" learning of Gaussians, where we are given samples from an arbitrary distribution and would like to find the best-fit mixture of k Gaussians. This problem naturally accounts for noise and appears to be much more realistic. Brubaker [Bru09] gave an algorithm that makes progress towards this goal, by allowing a mixture to be corrupted by an ϵ fraction of noisy points with $\epsilon < w_{\min}$, and with nearly the same separation requirements as in Section 2.2.3.

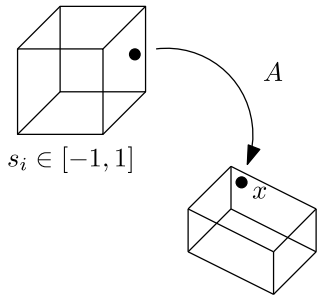
Chapter 3

Independent Component Analysis

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

$$x = As,$$

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



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

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

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

3.1 Recovery with fourth moment assumptions

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

Algorithm: ICA

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

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

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

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

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

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

Theorem 3.1. *Assume that*

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

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

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

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

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

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

Based on the assumptions about the signal,

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

which we can plug back into the previous equation.

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

Now, if we apply

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

We define the tensor

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

Then,

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

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

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

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

For any vector u,

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

We will pick a random Gaussian vector u. Then,

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

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

3.2 Fourier PCA and noisy ICA

Here we assume data is generated from the model

$$x = As + \eta,$$

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

Algorithm: Noisy ICA

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

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

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

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

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

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

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

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

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

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

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

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

Doing this for both u and v, we have

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

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

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

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

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

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

3.3 Discussion