Khoury College of Computer Sciences
Northeastern University
CS 7870: Seminar in Theoretical Computer Science

Fall 2023
6 & 10 October 2023
Scribe: Alma Mohadese Ghafari

Outline:

- Principal component Analysis
- Singular Value Decomposition
- The Power Iteration Method
- High-Dimensional Spaces: The Unit Ball

In this pair of lectures, we learn about the geometry of high-dimensional spaces and ways to address the complexity of high-dimensional data. In practice, many data sets that we would like to analyze are high-dimensional, which makes them difficult to analyze and reason about. For instance, we can visualize two- and three-dimensional data nicely, but it is very challenging to get an intuitive sense of data in higher dimensions. Furthermore, many data sets we work involve dimensions ranging in the thousands and more. Many geometric algorithms have running times exponential in the number of dimensions, rendering them unusable for high-dimensional spaces. One approach to address this is to embed or project the given data into a low-dimensional space, while having essentially the same features or desirable properties of the original data set. Of course, this is not always possible, but it turns out that when the underlying data satisfies some natural conditions that arise in practice, one can implement such dimensionality reduction techniques.

We begin by introducing two closely related methods from linear algebra—Principal Component Analysis and Singular Value Decomposition—that achieve part of this dimensionality reduction goal. We explore the relationship between these two approaches, and how they are related to the eigenvalues of the matrix representing the given data. We also present a randomized algorithm for approximating the top eigenvalue and associated eigenvector. Along the way, we explore the geometry of high dimensions and the properties of a unit vector chosen randomly in a high-dimensional Euclidean space.

The presentation of the material in this pair of lectures is heavily drawn from Chapters 2 and 3 of the text by Blum, Hopcroft, and Kannan [BHK20] and lecture notes of Roughgarden and Valiant [RV16b, RV16a].

1 Principal Component Analysis

Given a set of n points in d-dimensional space, the Principal Component Analysis (PCA) aims to find the best k-dimensional subspace (for $k \ll d$) that best captures the features of the set of points. Let us assume that we have n points $a_1, a_2, \ldots, a_n$ in $\mathbb{R}^d$. We represent these points using an $n \times d$ matrix $A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$. For convenience, throughout out study, we will assume that the mean of the point set is the origin; if this is not the case, then we can apply a simple translation to the points to make this assumption true. The goal of PCA

is to find a k dimensional subspace containing the origin that minimizes the sum of squares of perpendicular distances of each point to the subspace.

For any point, set t_i as the distance of point i to the subspace, and s_i be the projection length. By Pythagoras theorem, we have:

$$\min \sum_{i=1}^{n} t_i^2 = \max \sum_{i=1}^{n} (t_i^2 - a_i^2) = \max \sum_{i=1}^{n} s_i^2.$$

This, the objective of minimizing the sum of the squares of perpendicular distances of the points to the subspace is equivalent to that of maximizing the sum of the squares of the lengths of the projection of the points to the subspace. This implies that we want to find a (unit) vector v of size d that maximizes $v^T A^T A v$.

Let $X = A^T A$, note that X is a symmetric matrix and by lemma 1 we can write it in the form of $Q^T D Q$.

Lemma 1. Any symmetric matrix X of size $d \times d$ can be written as $Q^T D Q$, where D is a diagonal matrix with eigenvalues of X, and Q is the $d \times d$ matrix consisting of the d eigenvectors of X.

1.1 1-dimensional subspace

We first consider the case where $k = 1$. So, we want the line through the origin such that sum of the squares of the projections of the points to the line is maximized. Let us first solve the problem for a special case where X is a diagonal matrix and $\lambda_1 \geq \cdots \geq \lambda_d$ are the eigenvalues of A, $v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$, we have:

$$v^T D v = \sum_{i=1}^{d} \lambda_i v_i^2.$$

Since v is a unit vector, $\sum_i v_i^2 = 1$. Therefore, $v^T D v$ is maximized when $v_1 = 1$ and $v_i = 0$ for $i > 1$.

For general X, we can write $Q = \begin{bmatrix} \dots u_1 \dots \\ \dots u_2 \dots \\ \vdots \\ \dots u_d \dots \end{bmatrix}$. Note that u_i is the ith eigenvector of X, then we have:

$$v^T Q^T D Q v = \begin{bmatrix} u_1 \cdot v & u_2 \cdot v & \cdots & u_d \cdot v \end{bmatrix} D \begin{bmatrix} u_1 \cdot v \\ u_2 \cdot v \\ \vdots \\ u_d \cdot v \end{bmatrix} = \sum_{i=1}^{d} \lambda_i (u_i \cdot v)^2.$$

We thus obtain that if $v = u_1$, $v^T Q^T D Q v = \lambda_1 |u_1|^2 = \lambda_1$. Furthermore, for any v, we have $u_i \cdot v \leq |u_i||v|$. So, we have

$$v^T Q^T D Q v = \sum_{i=1}^{d} \lambda_i (u_i \cdot v)^2 \leq \sum_{i=1}^{d} \lambda_i |u_i|^2 |v|^2 = \sum_{i=1}^{d} \lambda_i |u_i|^2 \leq \lambda_1.$$

We have thus derived that line that maximizes the sum of the squares of the projections to the line is given by the first eigenvector of X.

1.2 Extending to k-dimensional subspaces

We now consider the case $k = 2$. Again, let us assume that X is a diagonal matrix. Our goal is to find vectors v and w to maximize $v^T Dv + w^T Dw = \sum_{i=1}^{d}(v_i^2 + w_i^2)\lambda_i$, subject to the condition that w is orthogonal to v. In this case we can prove that the objective is maximized if we set $v_1 = 1$ and $v_i = 0$ for $i > 1$ and $w_1 = 0, w_2 = 1$ and $w_i = 0$ for $i > 2$. In general, for k-dimensional subspaces, we have the following theorem.

Theorem 1. The k-dimensional subspace that maximizes the sum of projection length squares is formed by the top k eigenvectors of $A^T A$.

2 Singular Value Decomposition

An alternative way to answer the above question is to define a greedy iterative algorithm that finds vectors that maximize the sum of the projection length squares in orthogonal directions. Let v_1 be defined as follows.

$$v_1 = \arg\max_{v} |Av|$$

For $i > 1$, we define

$$v_i = \arg \max_{v_i \perp (v_1,\ldots,v_{i-1}) | v_i|=1} |Av|.$$

We stop the iterations at index r where $\max_{v\perp(v_1,\ldots,v_r)} |Av| = 0$. We call $v_1, \ldots, v_r$ left singular vectors of A. We define the singular values $\sigma_i = |Av_i|$ for all i, and finally we define the left singular values $u_i = \frac{1}{\sigma_i} Av_i$.

Theorem 2. The matrix A satisfies the following property.

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

This decomposition is called singular value decomposition and can be rewritten as $A = USV^T$ where

$$U^T = \begin{bmatrix} \ldots u_1 \ldots \\ \ldots u_2 \ldots \\ \vdots \\ \ldots u_r \ldots \end{bmatrix}, V^T = \begin{bmatrix} \ldots v_1^T \ldots \\ \ldots v_2^T \ldots \\ \vdots \\ \ldots v_r^T \ldots \end{bmatrix}$$

and S is a diagonal matrix with σ_i as the i-th diagonal.

Proof. We prove the above equality by showing that for any vector x we have Ax is the same as $USV^T x$. If $x \perp \{v_1, v_2, \ldots, v_r\}$, then we have $v_i^T x = 0$ resulting in $USV^T x = 0$. Note that by the greedy algorithm

introduced to find the left singular vectors, we have $Ax = 0$. For any vector x that is not perpendicular to v, we can write it as follows

$$x = \alpha_1 v_1 + \cdots + \alpha_r v_r,$$

by definition of singular values we know that $Av_i = \sigma_i u_i$ holds and therefore we can write

$$Ax = \sum_{i=1}^{r} \alpha_i \sigma_i u_i = USV^T V\alpha = USV^T x.$$

□

We now argue that SVD indeed yields the best fitting k-dimensional subspace.

Theorem 3. The first k vectors computed by the greedy algorithm defining the SVD yield a k-dimensional subspace that maximizes the sum of squared projections of the matrix A to the subspace.

Proof. The proof is by induction on k. The case $k = 1$ follows from the definition of the first vector. We now establish the induction step. Let V_i denote the i-dimensional subspace defined by the first i vectors v_1 through v_i. Suppose the claim is true for $k-1$. So, V_{k-1} is the $(k-1)$-dimensional space that maximizes the sum of squared projections of the matrix A to any $(k-1)$-dimensional subspace. Suppose W is the best k-dimensional subspace. We choose an orthonormal basic $w_1, w_2, \ldots, w_k$ so that w_k is perpendicular to v_1 through v_{k-1}. We can do this by projecting v_1 through v_{k-1} to W, and then selecting w_k perpendicular to all these projections. Then, we obtain

$$|Aw_1|^2 + |Aw_2|^2 + \ldots + |Aw_{k-1}|^2 \leq |Av_1|^2 + |Av_2|^2 + \ldots + |Av_{k-1}|^2$$

because V_{k-1} is the best k-dimensional subspace. Furthermore, among all vectors v orthogonal to v_1 through v_{k-1}, v_k maximizes $|Av|^2$, so $|Av_k|^2 \geq |Aw_k|^2$. Thus, we have

$$|Aw_1|^2 + |Aw_2|^2 + \ldots + |Aw_k|^2 \leq |Av_1|^2 + |Av_2|^2 + \ldots + |Av_k|^2,$$

completing the induction step and the proof of the theorem. □

3 Relation between PCA and SVD

Now, we explore the similarities of PCA and SVD, and how they both can be used to find a k-dimensional subspace that maximizes the sum of square projections of a matrix A to the subspace.

Theorem 4. The top k-eigenvectors of $A^T A$ form the k-dimensional subspace that that maximizes the sum of square projections of a matrix A to the subspace. Furthermore, this is the same as the top k right singular vectors of A created by SVD.

Proof. We sketched the argument for the first statement in our analysis of PCA. By our analysis above, we have $A^T A = QDQ^T$, and by SVD, we know how to decompose A, therefore here we have: $A^T A = VS^T UTUSV^T = VS^2 V^T = QDQ^T$. This implies that $V = Q$ and also the eigenvalues of $A^T A$ are the squares of the singular values: $\lambda_i = \sigma_i^2$ □

Note that both approaches above rely on determining the eigenvectors of a given matrix X. There are many algorithms for calculating the eigenvectors, many of which rely on the *power iteration method*, a fast randomized algorithm for finding the top eigenvector of a given matrix.

4 The Power Iteration Method

Here we introduce an algorithm to find the top eigenvectors of a given matrix X. Specifically, we present a method for finding the top eigenvector, which can then be iteratively used to find other top eigenvectors. Recall that the top eigenvector is the (unit) vector v that maximizes λ in the following

$$Xv = \lambda v$$

So, our aim is to a vector in which X has the highest projection. Let $\lambda_1 \geq \cdots \geq \lambda_n$ denote the eigenvalues and $v_1, \ldots, v_n$ the corresponding eigenvectors. We pick a unit vector $u_0 = \sum_{i=1}^{n} \alpha_i v_i$ at random, and define u_t as follows:

$$u_t = \frac{X^t u_0}{|X^t u_0|}$$

.

Theorem 5. Suppose v_1 is the top eigenvector for matrix A, then the following holds:

$$\lim_{t \to \infty} < u_t, v_1 >= 1$$

By the lemma below we prove the theorem 5 using the properties of a random unit vector.

Lemma 2. For a unit vector $u_0 = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$ chosen uniformly at random from the unit ball in n dimensions, with probability at least $\frac{1}{2}$ we have $\alpha_i > \frac{1}{2\sqrt{n}}$.

We now prove Theorem 5.

Proof. Let us first rewrite u_t as follows:

$$u_t = \frac{X^t u_0}{|X^t u_0|} = \frac{X^t [\sum_{i=1}^{n} \alpha_i \lambda_i^t v_i]}{|\sum_{i=1}^{n} \alpha_i \lambda_i^t v_i|}$$

$$< u_t, v_1 >= \frac{\alpha_1 \lambda_1^t}{\sqrt{\sum_{i=1}^{n} \alpha_i^2 \lambda_i^{2t}}} \geq \frac{\alpha_1 \lambda_1^t}{\sqrt{\alpha_1^2 \lambda_1^{2t} + \lambda_2^{2t}}}$$

By lemma 2 we have:

$$\frac{\alpha_1\lambda_1^t}{\sqrt{\alpha_1^2\lambda_1^{2t}+\lambda_2^{2t}}} = \frac{\alpha_1\lambda_1^t}{\alpha_1\lambda_1^t\sqrt{1+\frac{\lambda_2^{2t}}{\lambda_1^t}\frac{1}{\alpha_1^2}}} \geq 1-\frac{1}{2}\left(\frac{\lambda_2}{\lambda_1}\right)^{2t}\cdot 4n$$

If $\frac{\lambda_2}{\lambda_1} < 1$, setting $t = \log_{\frac{\lambda_1}{\lambda_2}} \frac{2n}{\epsilon}$ we get that $< u_t, v_1 >$ is at least $1-\epsilon$. Therefore, the number of iterations is $O(\ln(\frac{\lambda_1}{\lambda_2}n))$. And the convergence occurs unless λ_1 and λ_2 are equal. □

If λ_1 and λ_2 are well-separated, then the power iteration method essentially converges in logarithmic time. We will not formally prove lemma 2, but introduce the basics of high-dimensional geometry that help us reason about unit balls and spherical Gaussians, which are important concepts behind many modern algorithmic techniques.

5 High-dimensional spaces

5.1 Ball

We define a ball of radius r in d-dimension as the union of points with absolute value less than r the radius of the ball: $B(r) = \{x : |x| \leq r\}$.

5.2 Volume

We define the volume of a high-dimension ball, as the number of unit cubes inside the ball. Using calculus, one can derive the volume $V(d)$ and surface area $A(d)$ of a d-dimensional ball of unit radius (also referred to as a *unit ball*) as follows.

$$V(d) = \frac{2\pi^{\frac{d}{2}}}{d\Gamma(\frac{d}{2})} \quad A(d) = \frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})},$$

where the Gamma function $\Gamma(x)$ is a generalization of the factorial function to non-integers: $\Gamma(x) = (x-1)\Gamma(x-1), \Gamma(1) = \Gamma(2) = 1, \Gamma(\frac{1}{2}) = \sqrt{\pi}$.

Theorem 6. For a ball of radius r in d-dimensions, the fraction of the volume that is present in the sub-ball within radius $(1-\varepsilon)r$ is at most $e^{-\varepsilon d}$.

Proof. For any d-dimension ball of radius r we have

$$\text{vol}(B((1-\epsilon)r)) = (1-\epsilon)^d \text{vol}(B(r)) \leq e^{-\varepsilon d}\text{vol}(B(r)).$$

Note that if $\epsilon = \frac{k}{d}$ we have

$$\frac{vol(B((1-\epsilon)r)}{vol(B(r))} = e^{-k}$$

.

This means the volume of a unit ball is concentrated in an annulus of width $O(1/d)$ near the boundary. □

The following theorem states that the volume of a unit ball is concentrated around the equator. We omit the proof.

Theorem 7. For a unit ball in d-dimensions, at least $e^{-\frac{c^2}{2}}$ fraction of volume of a unit ball has $|x_1| \leq \frac{c}{\sqrt{d-1}}$.

The above theorem implies that if we set the north pole to be $(1, 0, \ldots, 0)$, most of the volume of the unit ball is concentrated in the equator corresponding to this north pole. Note that this statement holds for every coordinate. At the same time, we know that a significant volume of a unit ball in d-dimensions is on the surface. Therefore, if a pick a random unit vector, it holds that the typical value for each coordinate is of $\pm O\left(\frac{1}{\sqrt{d}}\right)$, which justifies Lemma 2.

Theorem 8. Suppose that we pick vectors $x_1, \ldots, x_n$ at random in $\mathbb{R}$. With probability $1 - \frac{1}{n}$ we have the following:

1. For any $i, j \in \{1, \ldots, n\}$such that $i \neq j$ we have $|x_i \cdot x_j| \leq \frac{\sqrt{6 \ln n}}{\sqrt{d-1}}$
2. For any $i \in \{1, \ldots, n\}$ we have $|x_i| \geq 1 - 2\frac{\ln n}{d}$

Proof. Here we calculate the above probabilities using theorem 7 and theorem 6.

1. Fix index i. We apply a rotation so that x_i is the along the line from the origin to the North Pole. Hence $|x_i \cdot x_j|$ is simply the magnitude of the projection of x_j in one coordinate. By theorem 7, we thus have $\Pr[|x_i \cdot x_j| \geq \frac{\sqrt{6 \ln n}}{\sqrt{d-1}}] \leq e^{-\frac{6 \ln n}{2}} = \frac{1}{n^3}$. Since there are at most $n^2/2$ pairs, the probability that the dot product for any pair is at least $\frac{\sqrt{6 \ln n}}{\sqrt{d-1}}$ is at most $1/(2n)$.
2. If $|x_i|$ is less than $1 - 2\frac{\ln n}{d}$, then it is within radius $1 - 2\frac{\ln n}{d}$ from the origin. By theorem 6, the probability of this occurring is

$$\left(1 - 2\frac{\ln n}{d}\right)^d \leq e^{-2 \ln n} = \frac{1}{n^2}.$$

   Thus, the probability that any vector has length at most $1 - 2\frac{\ln n}{d}$ is at most $1/n$.

□

References

[BHK20] Avrim Blum, John Hopcroft, and Ravi Kannan. *Foundations of Data Science*. Cambridge University Press, Cambridge, 2020.

[RV16a] T. Roughgarden and G. Valiant. How PCA works. Lecture notes, available from https://timroughgarden.org/s17/l/l8.pdf, 2016.

[RV16b] T. Roughgarden and G. Valiant. Understanding and using principal component analysis (PCA). Lecture notes, available from https://timroughgarden.org/s17/l/l7.pdf, 2016.