Mathematics of Machine Learning– Lecture Notes

Han-Miru Kim

March 18, 2021

Organisation

Professor: Dr. Afonso Bandeira

0 Introduction

Let's say we have some points $x_1, \ldots, x_n \in \mathbb{R}^d$. We ask what we can do with this data. In a statistics course, we might try to model the data, but that is not of interest here.

- One thing we can try is to reduce the dimension d to, say d'. And if you chose the right projection we might be able to find some structure in it. It turns out that this approach is mathematically rich.
- Another approach is to do **clustering** to find out about the geometry of the data.
- If the data consists of a graph with these nodes $x_1, \ldots, x_n$, then **Spectral Graph Theory** concerns itself with trying to cluster the nodes of the network.

One example we can look at is the so called **Netflix Problem**. Consider a matrix of ratings, where every uses has a row and the columns correspond to different movies. Since any given user only looks at a very small subset of all movies, the matrix will have a very low rank. What the company wants to know is what movies to suggest to a user. Can we "complete" the matrix if we know if it is low rank? How many entries do we know at least to meaningfully fill in the rest?
More generally, the problem of matrix completion is related to the problem of exploiting **sparsity** and **compresed sensing**.
The idea of data completion is apparent of supervised learning. Say we have some data set x_i which we know how to classify. Our goal is to obtain an agent that can correctly classify from new data. What differentiates "good" agents from "bad" agents is their ability to *generalize*. One might be more accurate when tested on the training data, but the other might do better when given *new* data.
Another topic of interest is that of **online learning**, where we want to learn from a continuous stream of data.
This has many applications in environments, which can change very fast. For example the stock market or Advertising.
There we want to continually optimize our agent. Some key concepts are gradient descent and backpropagation.
Say we have a set of points $S = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$ and we want to create k clusters.
What we want to minize is to find k means $a_1, \ldots, a_k$ and partition S into $S_1, \ldots, S_k$ to minimize the sum

$$\sum_{l=1}^{k} \sum_{i \in S_l} \|x_i - a_l\|^2$$

The problem is that this is a NP-hard problem, which means that we (currently) haven't found a fast algorithm to solve this problem.
There is however **LLoyd**'s Algorithm for k-means which is a popular approach, where we start with some initial data and then try to optimize the rest of the data to optimize the sum.

Proposition 0.0.1.
- *Given a choice for the centers $a_1, \dots, a_n$, the partition that minimizes the k-means objective is simply to assign x_k to the closest centers.*
- *Given a choice for the partition $S_1 \sqcup \dots \sqcup S_n$,the centers that minimize the above sum are given by*

$$a_k = \frac{1}{|S_k|} \sum_{i \in S_k} x_i$$

LLoyd's Algorithm goes a follows: We start with a partition at random and compute the best centers. Then we use these centers to create the best partitions. We then use these partitions to find the best centers etc. We can ask whether this proces terminates, or calculates the best outcome. Sadly, we don't always get the global minimum. There are also algorithmical problems as it is an NP-hard problem.
Also notice that the k-means problem is not the universal best solution to find a good clustering:

- One needs to know k beforehand and the points need to lie in $\mathbb{R}^d$, which is also not a sure given.
- There are also algorithmical problems as it is an NP-hard problem.
- One other problem is that our data needs to convex. If our data forms multiple rings around some point, we can intuitively cluster the data by these rings. But the k-means problem will never find this out. We can kind of solve this by re-mapping the points by convuluting them with a good kernel.

1 SVD

Often, data is represented by a matrix, say $X \in \mathbb{R}^{m \times n}$
We can think of X as a collection of n datapoints represented by a columns.
What is also possible is to think of each row and column as entities and the entry x_{ij} corresponds to the relationship by entity i and j.
A good way to better understand a matrix is through its **singular value decomposition**

$$X = U\Sigma V^T, \quad U \in O(m) V \in O(n), \quad \Sigma = \begin{pmatrix} \sigma_1 & \ddots & & 0 \\ & & \sigma_k & \\ 0 & & & 0 \end{pmatrix} \in \mathbb{R}^{m \times n}$$

where $\sigma_1 \geq \dots \geq \sigma_k$ and $k = \min\{m, n\}$. Recall that the columns $u_1, \dots, u_m$ of U and the columns $v_1, \dots, v_n$ of V are called the **left/right singular vectors**.
If $r = \operatorname{rank}(X)$, we can also write the SVD using slightly modified matrices

$$X = \tilde{U}\tilde{\Sigma}\tilde{V}^T, \quad \tilde{U} \in \mathbb{R}^{m \times r} \tilde{\Sigma} \in \mathbb{R}^{r \times r}, \tilde{V} \in \mathbb{R}^{n \times r}$$

where $U^T U = V^T V = 1_{r \times r}$ And to calculate this, we would for example do the following

(a) $\sigma_1^2 \geq \dots \sigma_r^2$ are the eigenvalues of $X^T X$ (or XX^T)

(b) $\tilde{U} = (u_1 | \dots | u_r)$ consisting of an ONB of $\mathbb{R}^m$ of Eigenvectors of AA^H

(c) $\tilde{V} = (v_1 | \dots | v_r)$ consisting of an ONB of $\mathbb{R}^m$ of Eigenvectors of $A^H A$

then we can write

$$X = \sum_{k=1}^{r} \sigma_k u_k v_k^T$$

what makes the alternative definition of the SVD a little bit nicer is that for $r < \min\{m, n\}$, the matrices are smaller.
A key idea and observation is that the data is often very well aproximated by a low rank matrix. If we were to plot the singular values σ_i with respect to i, then we very often see that their magnitude decreases quickly as i kets bigger.
We can make use of this by picking some $l < k$ and setting all σ_i to zero for $i > l$ and obtain a rank l matrix that *approximates* X.

Example 1.1 (See jupyter Notebook)**.** Let's take a look at an example, where we have an $m \times n$ pixel image and use the brightness to get the matrix entries. Then take a low rank approximation and plot the image of the low rank matrix aproximation. We were able to copmress an image to be only about 5% of its original size and get a pretty nice image back. It is of course missing some details, but it is nonetheless surprising that alot of the key features are still present.

It turns out that this particular way of approximating the data X by a low-rank matrix is *the "best"* way of doing it. But in order to properly define that the "best" is, we need a way to define distances between the data X and an approximation B.
For this, must introduce norms for matrices. A very natural norm is the **spectral-norm** defined by

$$\|A\|_2 := \max_{\substack{x \in \mathbb{R}^n \\ \|x\|=1}} \|Ax\|_2 = \sigma_1$$

The other very common norm is the **Frobenius norm**

$$\|A\|_F := \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\mathrm{tr}(A^T A)}$$

which are invariant under orthogonal transformations, i.e for $U, V \in U(n), U(m)$ we have

$$\|UAV\|_2 = \|A\|_2, \quad \|UAV\|_F = \|A\|_F = \sqrt{\sigma_1^2 + \ldots + \sigma_r^2}$$

and they are subadditive.
There is also a natural generalisation of these norms:

Definition 1.2 (Schatten p-Norm)**.** For $p \in [1, \infty]$ let

$$\|X\|_{s,p} = \sqrt[p]{\sum_{i=1}^{k} \sigma_i^p(x)} = \|\sigma(X)\|_p$$

where we see $\sigma(X)$ as a vector of the eigenvalues.

So now let

$$X_r := U_r \Sigma_r V_r, \quad U_r \in \mathbb{R}^{m \times r}, V_r \in \mathbb{R}^{n \times r}, \Sigma_r \in \mathbb{R}^{r \times r}$$

consist of the first r columns of their counterparts. It should be clear that

$$\mathrm{rank}(X_r) = r, \quad \sigma_1(X - X_r) = \sigma_{r+1}(X)$$

Lemma 1.3 (Eckart Young Mirsky for spectral norm)**.** *Let $X = UDV^H$ with singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k \geq 0$. And X_r defined as above. For any rank r matrix $B \in \mathbb{R}^{n\times m}$*

$$\|X - B\| \geq \|X - X_r\|$$

Proof. Let $X = U\Sigma V^T$ be the SVD of X. Since $\text{rank}(B) = r$. thre exists a non-zero vector in the span of $v_1, \ldots, v_{r+1}$ such that $Bw = 0$. Without loss of generality, we can scale w to get $\|w\| = 1$ and we can assume that $v_1, \ldots v_{r+1}$ are orthonormal. Then

$$w = \sum_{i=1}^{r+1} \alpha_i v_i \quad \text{for} \quad \alpha_i = v_i^T \omega, \quad \text{and} \quad \sum_{i=1}^{r+1} \alpha_i^2 = \|w\|^2 = 1$$

so using the fact that $V^T\omega$ consists of rows $v_i^T\omega = \alpha_i$ and that the v_i are orthonormal, we get

$$\begin{aligned}\|X - B\| &= \max_{\|v\|=1} \|(X - B)v\| \geq \|(X - B)\omega\| = \|X\omega\| = \|U\Sigma V^T\omega\| \\ &= \|\Sigma V^T\omega\| = \sqrt{\sum_{i=1}^{r+1} \sigma_i^2(X)\alpha_i^2} \geq \sigma_{r+1}(X) = \sigma_1(X - X_r) = \|X - X_r\|\end{aligned}$$

□

Theorem 1.4 (Weyl inequalities for singular values)**.**

$$\sigma_{i+j-1}(X + Y) \leq \sigma_i(X) + \sigma_j(Y)$$

for all $1 \leq i, j \leq \min\{m, n\}$ satisfying $i + j - 1 \leq \min\{m, n\}$

Proof. Let $X_{i-1}$and Y_{j-1} be the rank $i - 1$, $j - 1$ approximations of X and Y. By the triangle inequality of the spectral norm, we have

$$\sigma_1((X - X_{i-1}) + (Y - Y_{j-1})) \leq \sigma_1(X - X_{i-1}) + \sigma_1(Y - Y_{j-1}) = \sigma_i(X) + \sigma_j(Y)$$

and since $X_{i-1} + Y_{j-1}$ has at most rank $i + j - 2$, the Eckart Young Mirsky theorem gives us

$$\sigma_{i+j+1}(X + Y) = \sigma(X + Y - (X + Y)_{i+j-2}) \leq \sigma_1(X + Y - (X_{i-1} + Y_{j-1}))$$

Putting both inequalities together, the proof follows. □

There is also another proof using the Courant-Fischer minimax carachterisation of singular values.

Theorem 1.5 (Eckart Young Mirsky)**.** *The truncated SVD gives the best low-rank approximation in any Schatten p-norm. So for $X \in \mathbb{R}^{m\times n}$, $r < \min\{m, n\}$, $p \in [1, \infty]$, let X_r as above. Then for any rank r matrix B*

$$\|X - B\|_{s,p} \leq \|X - X_r\|_{s,p}$$

Proof. We already have proven this for $p = \infty$, so let $1 \leq p < \infty$. Using the Weyl inequality for $X - B$ and B we get

$$\sigma_{i+j-1}(X) \leq \sigma_i(X - B) + \sigma_j(B)$$

For $j = r+1$, we know that since B is rank r, $\sigma_{r+1}(B) = 0$, so for all $i > 1$ such that $i + (r+1) - 1 \leq \min\{m, n\}$ we have

$$\sigma_{i+r}(X) \leq \sigma_i(X - B) + 0$$

thus

$$\|X - B\|_{s,p}^p = \sum_{k=1}^{\min\{m,n\}} \sigma_k^p(X - B) \geq \sum_{k=1}^{\min\{m,n\}-r} \sigma_k^p(X - B) \geq \sum_{k=1}^{\min\{m,n\}-r} \sigma_{k+r}^p = \sum_{l=r+1}^{\min\{m,n\}} \sigma_{l+r}^p(X) = \|X - X_r\|_{s,p}^p$$

□

2 Dimension Reduction

The idea in dimension reduction (or principal component analysis) is that we have data consisting of m points $y_1, \ldots, y_m$ in a high-dimensional space $\mathbb{R}^p$, and we want to find the best d-dimensional approximation of the data, where $d \ll p$.
We start by centering the points around their weighted average. So let

$$x_k = y_k - \frac{1}{m} \sum_{k=1}^{m} y_k$$

and we are looking for m points $z_1, \ldots, z_m$ that lie in a d-dimensional subpsace of $\mathbb{R}^p$ that lie as close as possible. Write Z as a matrix, where the rows are $z_1, \ldots, z_m$ and the same for X.

$$Z = \begin{pmatrix} | & & | \\ z_1 & \ldots & z_m \\ | & & | \end{pmatrix} \quad \text{and} \quad X = \begin{pmatrix} | & & | \\ x_1 & \ldots & x_m \\ | & & | \end{pmatrix}$$

The requirement that $z_1, \ldots, z_m$ lie in a d-dimensional space is equivalent to saying that $\text{rank}(Z) \leq d$. And requiring that they approximate $x_1, \ldots, x_m$ as best as possible is to minimize

$$\sum_{k=1}^{m} \|x_k - z_k\|^2 = \|X - Z\|_F^2$$

the solution to this is just the truncated SVD of X, so we would find

$$Z = U\Sigma V^T, \quad \text{for} \quad U \in \mathbb{R}^{p \times d}, \Sigma \in \mathbb{R}^{d \times d}, V \in \mathbb{R}^{m \times d}$$

The k-th column of V^T (and thus also ΣV^T) can be viewed as a representation of x_k as a d-dimensional vector. So let's define

$$\beta_k = k\text{-th column of } \Sigma V^T, \quad k = 1, \ldots, m$$

By letting μ to be the weighted average of all y_k, we can write

$$\frac{1}{m-1} XX^T = \frac{1}{m-1} \sum_{k=1}^{m} (y_k - \mu)(y_k - \mu)^T$$

but also, using the SVD of X: $X = U_f \Sigma_f V_f^T$ we get

$$\begin{aligned}\frac{1}{m-1} XX^T &= \frac{1}{m-1} U_f \Sigma_F V_f^T \left(U_f \Sigma_f V_f^T\right)^T \\ &= \frac{1}{m-1} U_f \Sigma_f^2 U_f^T\end{aligned}$$

Not instead of XX^T let's consider $X^T X$. We can make sense of the entries of the matrix as

$$(X^T X)_{ij} = \langle x_i, x_j \rangle$$

then take the leading eigenvectors of $X^T X$ and use them as a basis.
Another concept is that of **Manifold learning**, where we trust small distances over large distances.
[Missing 30 mins]
Given an undirected graph. We can take its adjacency matrix A and see what the leading eigenvectors of A say about the network.

$$\lambda_2(L) = 0 \iff G \text{ is disconnected}$$

We have seen the theorem that given a Graph $G = (V, E)$ with adjacency matrix A

$$\lambda_2(L) = 0 \iff G \text{ is disconnected}$$

The proof for $\implies$ was given last week.
For the other way: if G is disconnected, there exists non-trivial S, S^c such that for $y = 1_S$, y and 1 are linearly independent, so their span is two-dimensional.
Then we can write

$$y^T L y = \sum_{(i,j)\in E} (y_i - y_j)^2 = 0$$

and because $L \geq 0$ and $Ly = 0$ and $1, y \in \text{Ker}(L)$, the dimension of the kernel must be ≥ 2. So $\lambda_2(L) = 0$.
Image a network with two clusters that are almost disconnected. Is it true that $\lambda_2(L)$ is almost 0?

Theorem 2.1. *Let $G = (V, E)$ without isolated nodes and let $\mathcal{L}_G$ be the normalized graph Laplacian*

$$\mathcal{L}_G = I - D^{-1/2} A D^{-1/2}$$

then, for cut(S) and Ncut(S) given by

$$cut(S) = \sum_{i\in S} \sum_{j\in S^c} 1_{(i,j)\in E}, \quad vol(A) = \sum_{x\in A} \deg(x)$$

$$\lambda_2(\mathcal{L}_G) \leq \min_{0\neq S\subset V} \left(\frac{cut(S)}{vol(S)} + \frac{cut(S)}{vol(S^c)} \right)$$

Proof. By the courant'fischer variational principle of eigenvalues, we have

$$\lambda_2(\mathcal{L}_G) = \sum_{\|z\|=1, z\perp v_1(\mathcal{L}_G)} z^T \mathcal{L}_G z$$

But we know that $v_1(\mathcal{L}_G) = D^{1/2}1$ so by a change of variables $z = D^{1/2}y$ we get

$$\lambda_2(\mathcal{L}_G) = \min_{y^T D^{1/2}\mathcal{L}_G D^{1/2} y, y^T D1 = 0} y^T D^{1/2}\mathcal{L}_G D^{1/2} y = \min_{y^T Dy = 1, y^T D1 = 0} \sum_{(i,j)\in E} (y_i - y_j)^2$$

By restricing the vectors y to only take two values for a set $S \subset V$

$$y_i = \begin{cases} a & \text{if } i \in S \\ b & \text{if } i \notin S \end{cases}$$

and we ask for which values of a and b does y satisfy the constraints?

$$y^T Dy = 1 \iff \sum_{i=1}^{n} y_i^2 \deg(i) = a^2 \sum_{i\in S} \deg(i) + b^2 \sum_{i \in S} \deg(i) = a^2 \text{vol}(S) + b^2 \text{vol}(S^c) = 1$$

Which requires the two following equations

$$a^2 \text{vol}(S) + b^2 \text{vol}(S^c) = 1 \quad \text{and} \quad \sum_{i=1}^{n} \deg(i) y_i = a\text{vol}(S) + b\text{vol}(S^c)$$

and we can show that the resulting y_i must be given by

$$y_i = \begin{cases} \left(\frac{\text{vol}(S^c)}{\text{vol}(S)\text{vol}(G)}\right)^{\frac{1}{2}} & \text{if } i \in S \\ -\left(\frac{\text{vol}(S)}{\text{vol}(S^c)\text{vol}(G)}\right)^{\frac{1}{2}} & \text{if } i \notin S \end{cases}$$

now we need to compute $y^T \mathcal{L}_G y$.

$$\begin{aligned} y^T \mathcal{L}_G y &= \sum_{(i,j)\in E} (y_i - y_j)^2 \\ &= \sum_{i\in S} \sum_{j \in S^c} 1_{(i,j)\in E} (y_i - y_j)^2 \\ &= [...] \\ &= \text{cut}(S) \frac{1}{\text{vol}(G)} \left[\frac{\text{vol}(S^c)}{\text{vol}S} + 2 + \frac{\text{vol}(S)}{\text{vol}(S^c)} \right] \\ &= \text{cut}(S) \left[\frac{1}{\text{vol}(S)} + \frac{1}{\text{vol}(S^c)} \right] = \text{Ncut}(S) \end{aligned}$$

□

Given a cluster of a graph, we want to find a partition S, S^c with small Ncut.