CSE 521: Design and Analysis of Algorithms I **Fall 2020**

Lecture 11: Low Rank Approximation

Lecturer: Shayan Oveis Gharan *11/04/2020*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

In this lecture we discuss theory and applications of singular value decomposition.

The ideas described in the previous sections are used in low-rank approximation theory which finds many applications in computer science. A famous recent example was the Netflix problem. We have a large dataset of users and many of them have provided ratings to many movies. But this ratings matrix obviously has several missing entries. The problem is to figure out, using this limited data, what movies to recommend to users. Under the (justifiable) assumption that this is a low-rank matrix,this is a matrix completion problem that falls in the category of low-rank approximation.

So, we may, for example, leave the unknown entries to be 0. Then, we can approximate the matrix with low rank matrix. Then, we can fill out the unknown entries with the entries of the estimated low rank matrix. This gives a heuristic for the matrix completion problem.

Formally, in the low rank approximation problem we are given a matrix M, we want to find another $\tilde{M}$ of rank k such that $\|M - \tilde{M}\|$ is as small as possible.

Recall the Johnson-Lindenstrauss dimension reduction theorem tells us that for any set of n points $P \in \mathbb{R}^m$, with high probability, we can map them using $\Gamma \in \mathbb{R}^{d\times m}$, to a $d = \mathcal{O}(\log n)/\epsilon^2$ dimensional space such that for any $x, y \in P$,

$$(1-\epsilon)\|x-y\|^2 \leq \|\Gamma(x) - \Gamma(y)\|^2 \leq (1+\epsilon)\|x-y\|^2$$

As clear from this context, the dimension reduction ideas are oblivious to the structure of the data. That is the Gaussian mapping that we defined does not look at the data point to construct the lower dimensional map. Because of that it may now help us to observe certain hidden structures in the data. As we will see in the next lecture, low rank approximation algorithms, chooses the low rank matrix by looking at the SVD of M. Because of that it typically can reveal many unknown hidden structures between the data points that M represent.

1.1 Best Low Rank Approximation

Theorem 1.1. *For any matrix $M \in \mathbb{R}^{m\times n}$*

$$\inf_{\text{rank}(\hat{M})=k} \|M - \hat{M}\|_2 = \sigma_{k+1}, \tag{1.1}$$

where the infimum is over all rank k matrices $\hat{M}$.

We did not discuss the proof of this theorem in class. We are including the proof here for the sake of completeness.

Proof. To proof 1.1, we need to prove two statements: (i) There exists a rank k matrix $\hat{M}_k$ such that $\|M - \hat{M}_k\|_2 = \sigma_{k+1}$; (ii) For any rank k matrix $\hat{M}_k$, $\|M - \hat{M}_k\| \geq \sigma_{k+1}$.

We start with part (i). We let

$$\hat{M}_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \tag{1.2}$$

By definition of M,

$$M - \hat{M}_k = \sum_{i=k+1}^{m} \sigma_i \mathbf{u}_i \mathbf{v}_i^T \rightarrow \|M - \hat{M}_k\|_2 = \sigma_{max}(\sum_{i=k+1}^{m} \sigma_i \mathbf{u}_i \mathbf{v}_i^T) = \sigma_{k+1} \tag{1.3}$$

Now, we prove part (ii). Let $\hat{M}_k$ be an arbitrary rank k matrix. The null space of the matrix M,

$$\text{NULL}(M) = \{\mathbf{x} : M\mathbf{x} = 0\} \tag{1.4}$$

is the set of vector that M maps to zero. Let null(M) be the dimension of the linear space NULL(M). It is a well-known fact that for any $M \in R^{m \times n}$,

$$\text{rank}(M) + \text{null}(M) = n. \tag{1.5}$$

It is easy to see this from the SVD decomposition; null is just n minus the number of nonzero singular values of M. Putting it differently, any vector orthogonal to the right singular vectors of M is in NULL(M). So, the above equality follows from the fact that M has rank(M) right singular vectors (with positive singular values). As an application, since $\hat{M}_k$ has rank k, we have

$$\text{null}(\hat{M}_k) = n - \text{rank}(\hat{M}_k) = n - k. \tag{1.6}$$

By the Rayleigh quotient, we obtain

$$\begin{aligned}
\sigma_{k+1}(M)^2 = \lambda_{k+1}(M^T M) &= \min_{S:(n-k)dims} \max_{\mathbf{x} \in S} \frac{\mathbf{x}^T M^T M \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \\
&\leq \max_{\mathbf{x} \in \text{NULL}(\hat{M}_k)} \frac{\mathbf{x}^T M^T M \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \\
&= \max_{\mathbf{x} \in \text{NULL}(\hat{M}_k)} \frac{\mathbf{x}^T (M - \hat{M}_k)^T (M - \hat{M}_k) \mathbf{x}}{\mathbf{x}^T \mathbf{x}} \\
&\leq \max_{\mathbf{x}} \frac{\mathbf{x}^T (M - \hat{M}_k)^T (M - \hat{M}_k) \mathbf{x}}{\mathbf{x}^T \mathbf{x}}
\end{aligned} \tag{1.7}$$

The first inequality uses the fact that NULL(M) is a $n-k$ dimensional linear space; so a special case of S being a $n-k$ dimensional linear space is $S = \text{NULL}(M)$. The second equality uses that $\hat{M}_k \mathbf{x} = 0$ for any $\mathbf{x} \in \text{NULL}(\hat{M}_k)$.

Now, we are done using another application of the Rayleigh quotient.

$$\max_{\mathbf{x}} \frac{\mathbf{x}^T (M - \hat{M}_k)^T (M - \hat{M}_k) \mathbf{x}}{\mathbf{x}^T \mathbf{x}} = \lambda_{\max}((M - \hat{M}_k)^T (M - \hat{M}_k)) = \sigma_{\max}(M - \hat{M}_k)^2. \tag{1.8}$$

The completes the proof of 1.1. □

Theorem 1.2. *For any matrix $M \in R^{m \times n}$ (with $m \leq n$) with singular values $\sigma_1 \leq \sigma_2 \leq ... \leq \sigma_m$*

$$\inf_{\hat{M}_k : \text{rank}(\hat{M}_k) = k} \|M - \hat{M}_k\|_F^2 = \sum_{i=k+1}^{m} \sigma_i^2 \tag{1.9}$$

Proof. Since $\hat{M}_k$ has rank k, we can assume columns of $\hat{M} \in \text{span}\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_k\}$ where $\{\mathbf{w}_1, ..., \mathbf{w}_k\}$ is a set of orthonormal vectors for the linear space of columns of $\hat{M}_k$. First, observe that

$$\|M - \hat{M}\|_F^2 = \sum_{i=1}^{k} \|M_i - \hat{M}_i\|^2.$$

Now, let us answer the following question: Choose a vector $v \in \text{span}\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$ that minimizes $\|M_i - v\|^2$? It is not hard to see that the optimum solution is the projection of M_i onto the subspace of $\mathbf{w}_i$'s, i.e., we must have

$$\hat{M}_i = \sum_{j=1}^{k} \langle M_i, \mathbf{w}_i \rangle \mathbf{w}_i. \tag{1.10}$$

To write the above equation in matrix form we need to use the notion of a *projection matrix*. Given a set of orthonormal vectors $\mathbf{w}_1, \ldots, \mathbf{w}_k$, a projection matrix Π is defined as

$$\sum_{i=1}^{k} \mathbf{w}_i \mathbf{w}_i^T.$$

A projection matrix projects any vector in $\mathbb{R}^n$ in the linear span of its vectors; in the above case we have

$$\Pi_k M_i = \sum_{j=1}^{k} \langle \mathbf{w}_j, M_i \rangle \mathbf{w}_j.$$

Projection matrices have many nice properties that come in handy when we need to use them. For example, all their singular values are either 0 or 1. For any projection matrix Π, we have $\Pi^2 = \Pi$. There is a unique projection matrix of rank n which is the identity matrix.

Have this in hand, we can write $\hat{M} = \Pi_k M$. Therefore,

$$\|M - \hat{M}\|_F^2 = \|M - \Pi_k M\|_F^2 = \|(I - \Pi_k)M\|_F^2 = \|\Pi_k^{\perp} M\|_F^2$$

In the last identity $\Pi_k^{\perp}$ is nothing but the projection matrix on the space orthogonal to $\{\mathbf{w}_1, \ldots, \mathbf{w}_k\}$. More precisely, let us add $\mathbf{w}_{k+1}, \ldots, \mathbf{w}_n$ such that $\mathbf{w}_1, \ldots, \mathbf{w}_n$ form an orthonormal basis of $\mathbb{R}^n$. Then,

$$\Pi_k^{\perp} = \sum_{i=k+1}^{n} \mathbf{w}_i \mathbf{w}_i^T.$$

Now, we need to show that to minimize $\|\Pi_k^{\perp} M\|_F^2$, $\mathbf{w}_1, \ldots, \mathbf{w}_k$ must completely lie on the space of top singular vectors of M. Firstly, by (**??**),

$$\|\Pi_k^{\perp} M\|_F^2 = \text{Tr}(M^T {\Pi_k^{\perp}}^T \Pi_k^{\perp} M) = \text{Tr}(M^T \Pi_k^{\perp} M) = \text{Tr}(M M^T \Pi_k^{\perp})$$

The last identity follows by the invariance of the trace under cyclic permutation. Now, we can rewrite the above as follows:

$$\begin{aligned}
\text{Tr}(M M^T \Pi_k^{\perp}) &= \text{Tr}(\sum_{i=1}^{n} \sigma_i^2 \mathbf{u}_i \mathbf{u}_i^T \sum_{j=k+1}^{n} \mathbf{w}_j \mathbf{w}_j^T) \\
&= \sum_{i=1}^{n} \text{Tr}(\sigma_i^2 \mathbf{u}_i \mathbf{u}_i^T \mathbf{w}_j \mathbf{w}_j^T) \\
&= \sum_{i=1}^{k} \text{Tr}(\sigma_i^2 \langle \mathbf{u}_i, \mathbf{w}_j \rangle^2) \\
&= \sum_{i,j} \sigma_i^2 \langle \mathbf{u}_i, \mathbf{w}_j \rangle^2 \geq \sum_{i=k+1}^{n} \sigma_i^2
\end{aligned} \tag{1.11}$$

The second identity follows by linearity of trace and the third identity follows by invariance of the trace under cyclic permutation. To see that last inequality, observe that for any $k+1 \le j \le n$, $\sum_{i=1}^{n} \langle u_i, \mathbf{w}_j \rangle^2 = 1$. So, $\sum_{i,j} \sigma_i^2 \langle u_i, \mathbf{w}_j \rangle^2$ is minimized if all $\mathbf{w}_j$'s lie in the linear space $\text{span}\{\mathbf{u}_{k+1}, \ldots, \mathbf{u}_n\}$ as desired. □

The above theorems may not be desirable in practice because SVD computation is very costly. However, there are numerous algorithms that efficiently (in almost linear time) approximate the SVD of a given matrix. The following famous paper received the best paper award of STOC 2013 to resolve this question.

Theorem 1.3 (Clarkson-Woodruff 2013 [1])**.** *There is an algorithm that for any matrix $M \in \mathbb{R}^{n \times n}$ and an input parameter k, finds a rank k matrix $\hat{M}$ such that*

$$\|M - \hat{M}\|_F^2 \le (1+\epsilon) \inf_{N:\text{rank}(N)=k} \|M - N\|_F^2.$$

The algorithm runs in time $O(\text{nnz}(A) \cdot (\frac{k}{\epsilon} + k \log k) + n \cdot \text{poly}(\frac{k}{\epsilon}))$. Here $\text{nnz}(M)$ denotes the number of nonzero entries of M, i.e., the input length of M.

1.2 Applications

We conclude this lecture by giving several other applications of low rank approximation.

Hidden Partition Given a graph, assume that there are two hidden communities A, B each of size $n/2$ such that for any pair of individuals $i, j \in A$ there is an edge between them with probability p; similarly, for any pair of individuals $i, j \in B$ there is an edge with probability p. For any pair of individuals in the two sides the probability of existing an edge is q. Given a sample of such a graph and assuming $p \gg q$, we want to approximately recover A, B. If we reorder the vertices such that the first community consists of nodes $1, \ldots, n/2$ and the second one consists of $n/2+1, \ldots, n$, the expected matrix looks like the following:

$$\hat{M}_2 = \begin{bmatrix} p & q \\ q & p \end{bmatrix}$$

Observe that the above matrix is a rank 2 matrix. So, one may expect that by applying a rank 2 approximation of the adjacency matrix of the given graph he can recover the hidden partition. This is indeed possible assuming p is sufficiently larger than q.

Max-cut Our most formal application of low rank approximation. We use this idea to design an optimization algorithm for the maximum cut problem. Given a graph $G = (V, E)$ we want to find a set $S \subseteq V$ which maximizes $|E(S, \overline{S})|$. Although the min-cut can be solved optimally, max-cut problem is an NP-hard problem. The best known approximation algorithm for this problem has an approximation factor of 0.878 by a seminal work of Goemans and Williamson. They showed taht there is a polynomial time algorithm which always return a cut $(T, \bar{T})$ such that

$$|E(T, \bar{T})| \ge 0.878 \max_S |E(S, \bar{S})|. \tag{1.12}$$

Firstly, we formulate the max-cut problem algebraically. For a set $S \subseteq V$, let

$$\mathbf{1}_i^S = \begin{cases} 1 & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases} \tag{1.13}$$

be the indicator vector of the set S. We claim that for any set $S \subseteq V$,

$$|E(S, \bar{S})| = \mathbf{1}^{S^T} A \mathbf{1}^{\bar{S}}. \tag{1.14}$$

In fact, for any two vectors $\mathbf{x}$ and $\mathbf{y}$,

$$\mathbf{x}^T A \mathbf{y} = \sum_{i,j} x_i A_{i,j} y_j. \tag{1.15}$$

So we can rewrite the max-cut problem as the following algebraic problem:

$$\max_S |E(S, \bar{S})| = \max_{x \in \{0,1\}^n} \mathbf{x}^T A(\mathbf{1} - \mathbf{x}) \tag{1.16}$$

The rest of the proof is a fairly general argument and use no combinatorial structure of the underlying problem. Namely we give an algorithm to solve the quadratic optimization problem $\max_{\mathbf{x} \in \{0,1\}^n} \mathbf{x} A(\mathbf{1} - \mathbf{x})$ for a given matrix A.

We do this task in two steps. First, we approximate A by a low rank matrix and we show that the optimum solution of the optimization problem for the low rank matrix is close to the optimum solution of A. Then, we design an algorithm to approximately solve the above quadratic problem for a low rank matrix. Roughly speaking, using the low rank property we reduce our n dimensional problem to a k dimensional question, and then we use the k-dimensional question using an ϵ-net.

1.3 Reference

[1] Clarkson, Kenneth L., and David P. Woodruff. "Low rank approximation and regression in input sparsity time." In Proceedings of the forty-fifth annual ACM symposium on Theory of computing, pp. 81-90. ACM, 2013.

[2] McSherry, Frank. "Spectral partitioning of random graphs." In Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on, pp. 529-537. IEEE, 2001.

[3] Clarkson, Kenneth L., and David P. Woodruff. "Low rank approximation and regression in input sparsity time." In Proceedings of the forty-fifth annual ACM symposium on Theory of computing, pp. 81-90. ACM, 2013.