Advanced Algorithms

Lecture 18: Low Rank Approximation

Lecturer: Nathan Klein

1 Singular Value Decomposition

Remember that that *rank* of a matrix $A \in \mathbb{R}^{m\times n}$ is equal to to any of the following equivalent things:

1. The number of linearly independent rows of A.
2. The number of linearly independent columns of A.
3. The number of non-zero eigenvalues of A if $m = n$, or the number of non-zero singular values of A if $m \neq n$.

If a matrix is rank ℓ, then it can be written as a sum of ℓ rank 1 matrices. For any rank 1 matrix $A \in \mathbb{R}^{m\times n}$, we can write $A = uv^T$ for $u \in \mathbb{R}^m, v \in \mathbb{R}^n$.

Notice that eigenvalues are only defined for matrices $A \in \mathbb{R}^{n\times n}$. If $A \in \mathbb{R}^{m\times n}$ for $m \neq n$, then Ax cannot possibly equal λx because $Ax \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$. For matrices with $m \neq n$, the analog of an eigenvalue is a *singular value*.

Before we talk more about singular values, recall the spectral theorem for symmetric matrices we introduced earlier in the course:

Theorem 1.1 (Spectral Theorem)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. Then, there are n eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ with corresponding orthonormal eigenvectors $v_1, \ldots, v_n$ so that*

$$A = \sum_{i=1}^{n} \lambda_i v_i v_i^T = V\Lambda V^T$$

where V has columns $v_1, \ldots, v_n$ (so $V^T V = I$) and Λ is the diagonal matrix with $\Lambda_{ii} = \lambda_i$.

A natural question is: what about for general matrices $A \in \mathbb{R}^{m\times n}$, or for matrices in $\mathbb{R}^{n\times n}$ which are not symmetric? It turns out all you need to do is turn A into a symmetric matrix using $A^T A$, and then apply the spectral theorem.

Theorem 1.2 (Singular Value Decomposition (SVD))**.** *Let $A \in \mathbb{R}^{m\times n}$. Then there are orthonormal vectors $u_1, \ldots, u_\ell$, orthonormal vectors $v_1, \ldots, v_\ell$, and singular values $\sigma_1 \geq \sigma_2 \geq \ldots, \geq \sigma_\ell$ such that:*

$$A = \sum_{i=1}^{\ell} \sigma_i u_i v_i^T = U\Sigma V^T$$

In addition, we have $\ell \leq \min\{m, n\}$ and $\sigma_i \in \mathbb{R}_{>0}$ for all i. (And here U is the matrix with columns $u_1, \ldots, u_\ell$ and V the matrix with columns $v_1, \ldots, v_\ell$.)

The u_i are called the left singular vectors and the v_i the right singular vectors. There are a few differences in this statement compared to the spectral theorem. The most important difference is that instead of $v_i v_i^T$, we have $u_i v_i^T$, i.e., these vectors can differ. Second, we typically list the singular values in decreasing order, whereas eigenvalues are in increasing order. Finally, singular values are all positive: we can negate u_i to flip the sign of the singular value, and then by convention we just delete the singular values of value 0.

To get a handle on why things are a bit different, consider the following matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

This is a rank 1 matrix. But it's clear you can't write it in the form $\lambda v v^T$. So, we need to relax the criteria: we need to write it in the form $\lambda u v^T$, which is easy: just pick $u = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$ and $v = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$.

Now that we have a handle on it, let's prove the SVD. The nice thing about the SVD is that it gives us an ordering of how important each rank 1 matrix is: the bigger the value σ_i, the higher the contribution.

Proof of SVD. $A^T A \in \mathbb{R}^{n \times n}$ is PSD since by definition it has a square root. Since it is symmetric, we can apply the spectral theorem, so there are non-negative eigenvalues $\lambda_1, \ldots, \lambda_n$ and orthonormal $v_1, \ldots, v_n$ such that $A^T A = \sum_{i=1}^n \lambda_i v_i v_i^T$. First, throw away all λ_i which are 0 so that $A^T A = \sum_{i=1}^\ell \lambda_i v_i v_i^T$ after re-indexing for some $\ell \leq \min\{n, m\}$ since the rank of the matrix is at most $\min\{m, n\}$. Let $v_1, \ldots, v_\ell$ be the orthonormal set above. First, notice that

$$\|Av_i\|_2^2 = v_i^T A^T A v_i = v_i^T \lambda v_i = \lambda$$

since v_i is an eigenvector of $A^T A$. So, $\|Av_i\|_2 = \sqrt{\lambda}$. Now, define for all i:

$$u_i = \frac{Av_i}{\|Av_i\|_2} = \frac{Av_i}{\sqrt{\lambda_i}} = \frac{Av_i}{\sigma_i}$$

In other words, set $\sigma_i = \sqrt{\lambda_i}$. By definition, these vectors have norm 1. Furthermore, they are orthonormal, because when $i \neq j$ we have:

$$u_i^T u_j = \frac{(Av_i)^T}{\sigma_i} \frac{Av_j}{\sigma_j} = \frac{v_i^T A^T A v_j}{\sigma_i \sigma_j} = \frac{v_i^T \lambda_j v_j}{\sigma_i \sigma_j} = 0$$

where we used that the v_i are orthonormal. So, the only thing remaining to prove is that $A = \sum_{i=1}^\ell \sigma_i u_i v_i^T$. It suffices to prove that $Av_j = (\sum_{i=1}^\ell \sigma_i u_i v_i^T) v_j$ for all $1 \leq j \leq n$ for the basis $v_1, \ldots, v_n$. This is enough, because then:

$$Ax = A(\sum_{j=1}^n v_j \langle x, v_j \rangle) = \sum_{j=1}^n Av_j \langle x, v_j \rangle = \sum_{j=1}^n \sum_{i=1}^\ell (\sigma_i u_i v_i^T) v_j \langle x, v_j \rangle = \sum_{i=1}^\ell \sigma_i u_i v_i^T \sum_{j=1}^n v_j \langle x, v_j \rangle$$

and this is $(\sum_{i=1}^{\ell} \sigma_i u_i v_i^T)x$. So, since they have the same product with every vector, they are the same matrix. So let's prove that $Av_j = (\sum_{i=1}^{\ell} \sigma_i u_i v_i^T)v_i$ for all $1 \le j \le n$ for the basis $v_1, \ldots, v_n$.

$$(\sum_{i=1}^{\ell} \sigma_i u_i v_i^T)v_j = \sum_{i=1}^{\ell} \sigma_i u_i \langle v_i, v_j \rangle = \sigma_j u_j v_j^T v_j = \sigma_j u_j = \sigma_j \frac{Av_j}{\sigma_j} = Av_j \qquad \square$$

2 Low Rank Approximation

A common task is to take a matrix $A \in \mathbb{R}^{m \times n}$ and find a new *low rank* matrix that approximates A. This has many applications:

1. The most tangible application is image compression. Given a matrix of pixels, we can use a low-rank approximation to compress the image.

2. In recommendation systems, we have a matrix encoding user ratings. For example, suppose each row is a user, and each column is a movie. The entry is 0 if the user has not rated the movie, and otherwise is some integer rating, perhaps 1 if they liked it and -1 otherwise. Now, we want to figure out what movies a particular user will like. It turns out a pretty good approach here is to find a low rank approximation of this matrix. For example, if everyone likes exactly the same movies (all movies are just good or bad), that's a rank 1 matrix since every row is the same. If that is the ground truth, then a rank 1 approximation is likely to figure this out. In general, you could hope that there are only k features of every user that will generate their movie ratings, which would demonstrate that a rank k approximation exists.

Unsurprisingly, the best way to approximate a matrix A with a matrix of rank k is to use the top k singular values. There is even a theorem:

Theorem 2.1. *Let $A \in \mathbb{R}^{m \times n}$. Then, an optimal rank k approximation of A can be obtained by taking the top k singular values of A, i.e., $\tilde{A} = \sum_{i=1}^{k} \sigma_i u_i v_i^T$. Formally:*

$$\inf_{rank(\tilde{A})=k} \|A - \tilde{A}\|_2 = \sigma_{k+1}$$

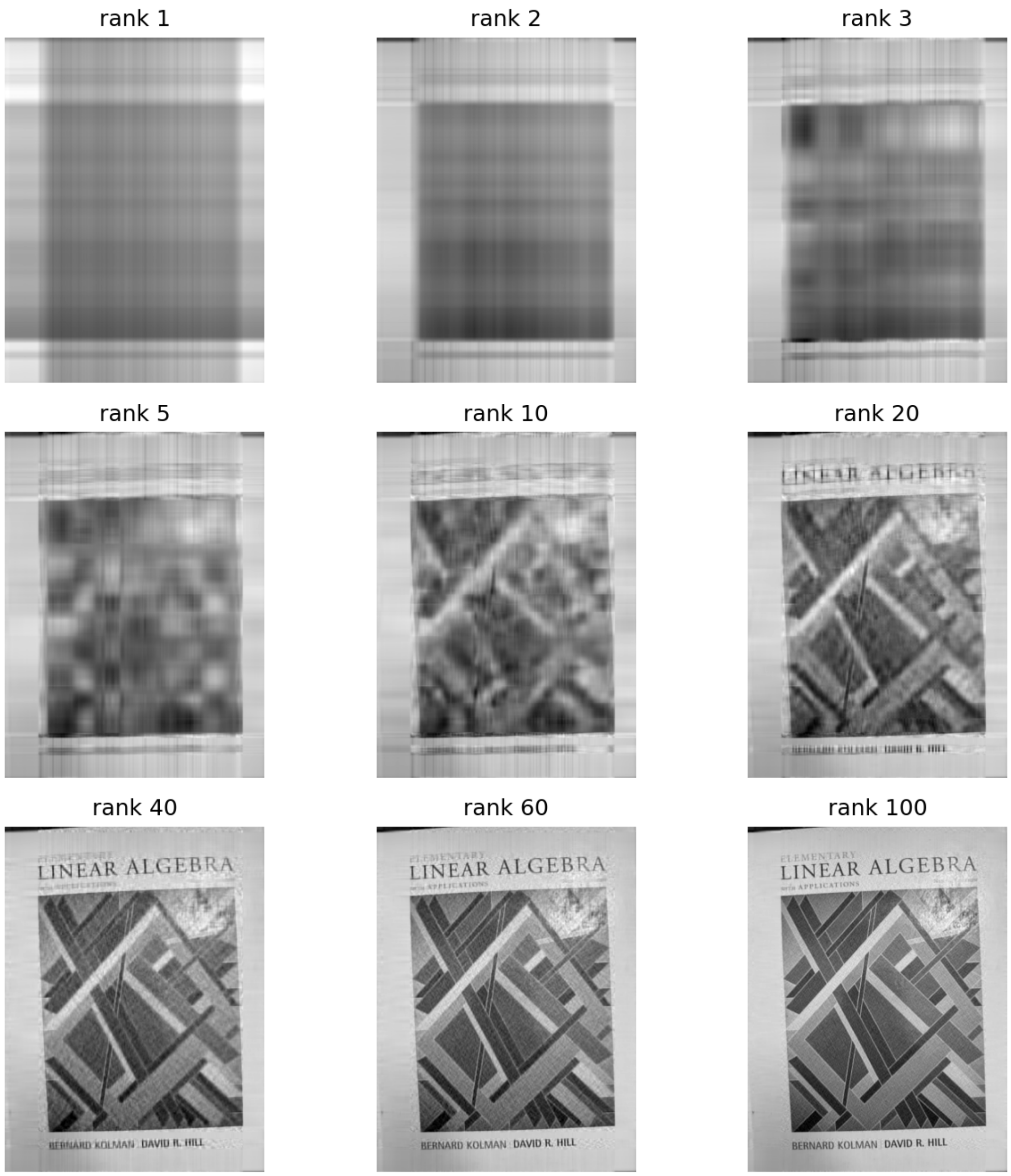
where recall $\|A\|_2$ for a matrix A is the spectral norm, or its largest singular value.

We will prove that $\|A - \tilde{A}\| \le \sigma_{k+1}$ by showing $\tilde{A} = \sum_{i=1}^{k} \sigma_i u_i v_i^T$ achieves this. We will leave the other direction as an exercise. Notice that:

$$A - \tilde{A} = \sum_{i=k+1}^{\ell} \sigma_i u_i v_i^T$$

So, $\|A - \tilde{A}\|_2 = \sigma_{max}(\sum_{i=k+1}^{\ell} \sigma_i u_i v_i^T) = \sigma_{k+1}$.

On your homework, you will visualize the effects of low rank approximation on images to produce something like the below. Naïvely, this is an image of about 1000×1000 pixels, which would require about half a megabyte if every pixel has 32 different possible values. To store a rank 60 version would require about a tenth of that.



3 Max Cut on Dense Graphs

A nice application of low rank approximation is a PTAS for the Max Cut problem on graphs with $\Omega(n^2)$ edges. When we discussed average case analysis, recall we saw that:

Fact 3.1. *Given a graph $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$, for a set $S \subseteq V$, let $\mathbf{1}_S \in \{0,1\}^n$ be the indicator vector of S. Let $A \in \{0,1\}^{n \times n}$ be the adjacency matrix of G so that $A_{ij} = 1$ if $\{v_i, v_j\} \in E$. Then,*

$$\mathbf{1}_S^T A \mathbf{1}_{V \setminus S} = |\delta(S)|$$

Max Cut is APX-Hard, so in general there is no PTAS for it. However, here we will show how the SVD leads to a PTAS in *dense* graphs, where there are at least δn^2 edges for some constant

$\delta > 0$. The idea is simple: first, find the best rank k approximation of the adjacency matrix A. Then, we will show that for all cuts S, we have:

$$\mathbf{1}_S^T A_k \mathbf{1}_{V \setminus S} \approx \mathbf{1}_S^T A \mathbf{1}_{V \setminus S}$$

with a maximum error on the order of $\frac{n^2}{\sqrt{k}}$. Then, we will show that there is an algorithm for solving max cut *optimally* on matrices of constant rank, one with run-time $k^{O(k)} \cdot \text{poly}(n)$. Together this gives a PTAS, since for any δ, ϵ we can choose k large enough so that $\frac{n^2}{\sqrt{k}} \leq \epsilon \cdot OPT$. Crucially, k only needs to depend on these constants δ, ϵ. (Note it is also OK for δ to be mildly sub-constant, but for simplicity we will assume it's a constant here.)

We first need the following, which we will use without proof:

Definition 3.2. *For any matrix A, the Frobenius norm $\|A\|_F$ is equal to:*

$$\|A\|_F = \sqrt{\sum_{i=1}^{n} \sigma_i^2} = \sqrt{\sum_{i,j} A_{ij}^2}$$

Lemma 3.3. *Let A_k be the best rank k approximation of a symmetric matrix $A \in \{0,1\}^{n \times n}$. Then for any $x \in \{0,1\}^n$,*

$$|x^T A(\mathbf{1} - x) - x^T A_k(\mathbf{1} - x)| \leq \frac{n^2}{\sqrt{k}}$$

Proof. In the lecture on average case analysis, we also proved that for any $x, y \in \mathbb{R}^n$ and symmetric matrix A, we have:

$$|x^T Ay| \leq \|x\|_2 \cdot \|y\|_2 \cdot \|A\|_2$$

Therefore, we have:

$$|x^T A(\mathbf{1} - x) - x^T A_k(\mathbf{1} - x)| = |x^T(A - A_k)(\mathbf{1} - x)| \leq \|x\|_2 \cdot \|\mathbf{1} - x\|_2 \cdot \|A - A_k\|_2$$

But we know that $\|A - A_k\|_2 = \sigma_{k+1}$ by Theorem 2.1. Furthermore, $\|x\|_2 \leq \sqrt{n}$ and similarly $\|\mathbf{1} - x\|_2 \leq \sqrt{n}$. So, we can upper bound this quantity by $n \cdot \sigma_{k+1}$.

But by the definition of the Frobenius norm, we have $\sum_{i=1}^{n} \sigma_i^2 = \|A\|_F^2 = \sum_{i,j} A_{ij}^2 \leq n^2$ since $A \in \{0,1\}^n$. So, the sum of the squares of the singular values is at most n^2. Therefore, the square of the $(k+1)$st largest one, σ_{k+1}, is at most $\frac{n^2}{k}$, so $\sigma_{k+1} \leq \frac{n}{\sqrt{k}}$. □

As a consequence, **we can approximate** A **with** A_k and not change the value of any cut by more than a $\pm \frac{n^2}{\sqrt{k}}$ factor. Using that the max cut is always at least $|E|/2$ and we have a dense graph with at least δn^2 edges, we can choose k so that this is at most an ϵ fraction of OPT. Then it is sufficient to solve the problem on A_k.

3.1 Solving Max Cut on Low Rank Matrices

Given A_k, we know that it is equal to $\sum_{i=1}^{k} \lambda v_i v_i^T$ (as A is symmetric, we can apply the spectral theorem here instead of the SVD, but the same ideas would work using SVD). But now given a vector x, $x^T A_k(\mathbf{1} - x)$, we have

$$x^T A_k(\mathbf{1} - x) = x^T(\sum_{i=1}^{k} \lambda_i v_i v_i^T)(\mathbf{1} - x) = \sum_{i=1}^{k} \lambda_i x^T v_i v_i^T(\mathbf{1} - x)$$

So, $x^T A_k(\mathbf{1} - x)$ is only a function of $\langle v_i, x\rangle$ for all i. There are only k of these numbers. So the idea is: let's *guess* what each inner product $\langle v_i, x\rangle$ should be! And we won't need to do so exactly: only up to some γ precision.

So, let's guess. For each possible inner product $\langle v_i, x\rangle$, it takes value between $-\sqrt{n}$ and $\sqrt{n}$ as $|\langle v_i, x\rangle| \leq \|v_i\|_2 \|x\|_2 \leq \sqrt{n}$ by Cauchy-Schwarz, since $\|v_i\|_2 = 1$. Now discretize the value of $\langle v_i, x\rangle$ into $[-\sqrt{n}, \sqrt{n}]$ using $2k$ equally spaced points $-\sqrt{n}, -\sqrt{n} + \frac{\sqrt{n}}{k}, \ldots, \sqrt{n} - \frac{\sqrt{n}}{k}, \sqrt{n}$.

Finally, let's brute force over all possible choices for $\langle v_i, x\rangle$ for all $1 \leq i \leq k$. There are $(2k)^k$ such choices. For each one, we get some value of $x^T A(\mathbf{1} - x)$, assuming it is the case that $\langle v_i, x\rangle$ equals the prescribed value. Our $\langle v_i, x\rangle$ have an error of at most $\frac{\sqrt{n}}{k}$ from optimal, and it turns out it's not hard to show this leads to an overall error of at most $O(\frac{n^2}{k})$.

What remains, then, is to show how to construct an actual $x \in \{0,1\}^n$ from these arbitrary numbers $\alpha_i = \langle v_i, x\rangle$. First, let's show some vector x exists: that's immediate.

Fact 3.4. *Let $x = \sum_{i=1}^k \alpha_i v_i$. Then, $\langle v_i, x\rangle = \alpha_i$ for all i.*

Proof.

$$v_i^T x = v_i^T \sum_{j=1}^k \alpha_j v_j = \alpha_i v_i^T v_i = \alpha_i$$

since the vectors v_i have length 1. □

So, we can try all such vectors x for our discretized space, and find the best one, and it will have value within a small ϵ factor of the optimal cut. Now we will take this optimal x. While it is not in $\{0,1\}^n$, it turns out we can produce a cut with it that does not lose objective value according to A_k, which is all we needed. We won't show how to do this in the lecture, but may see a related question on the homework.