Advanced Linear Algebra – Week 10

Applications of the Singular Value Decomposition

This week we will learn about:

- The pseudoinverse of a matrix,
- The operator norm of a matrix, and
- Low-rank approximation and image compression.

Extra reading and watching:

- Section 2.3.3 and 2.C in the textbook
- Lecture videos 38, 39, 40, and 41 on YouTube
- Moore–Penrose inverse (pseudoinverse) at Wikipedia
- Operator norm at Wikipedia
- Low-rank approximation at Wikipedia

Extra textbook problems:

⋆ 2.3.2, 2.3.4(d,e,h), 2.C.4(a,b,d,e)

⋆⋆ 2.3.8–2.3.12, 2.C.1–2.C.3

⋆⋆⋆ 2.3.15, 2.3.21, 2.C.5, 2.C.6, 2.C.9, 2.C.10

☠ 2.3.17(a)

The Pseudoinverse

We have been working with the inverse of a matrix since early-on in introductory linear algebra, and while we can do great things with it, it has some deficiencies as well. For example, we know that if a matrix $A \in \mathcal{M}_n$ is invertible, then the linear system $A\mathbf{x} = \mathbf{b}$...

However, that linear system might have a solution even if A is *not* invertible. For example...

Example. *Show that the linear system*

$$\begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 6 \end{bmatrix}$$

has a solution, but its coefficient matrix is not invertible.

It thus seems natural to ask whether or not there exists a matrix $A^\dagger$ with the property that a solution to the linear system $A\mathbf{x} = \mathbf{b}$ (when it exists) is $\mathbf{x} = A^\dagger\mathbf{b}$. Well...

Definition 10.1 — Pseudoinverse of a Matrix

Suppose $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$, and $A \in \mathcal{M}_{m,n}(\mathbb{F})$ has orthogonal rank-one sum decomposition

Then the **pseudoinverse** of A, denoted by $A^\dagger \in \mathcal{M}_{n,m}(\mathbb{F})$, is the matrix

There are several aspects of the pseudoinverse that we should clarify:

- If A is invertible, ...
- If A has SVD $A = U\Sigma V^*$, then...
- The pseudoinverse is well-defined.

Example. *Compute the pseudoinverse of the matrix*

$$A = \begin{bmatrix} 1 & 1 & 1 & -1 \\ 0 & 1 & 1 & 0 \\ -1 & 1 & 1 & 1 \end{bmatrix}.$$

The nice thing about the pseudoinverse is that it always exists (even if A is not invertible, or not even square), and it always finds a solution to the corresponding linear system (if a solution exists). Not only that, but if there are multiple different solutions, it finds the smallest one:

Theorem 10.1 — Pseudoinverses Solve Linear Systems

Suppose $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$, $A \in \mathcal{M}_{m,n}(\mathbb{F})$, and suppose that the system of linear equations $A\mathbf{x} = \mathbf{b}$ has at least one solution. Then

is a solution. Furthermore, if $\mathbf{y}$ is any other solution then $\|A^\dagger \mathbf{b}\| < \|\mathbf{y}\|$.

Proof. We start by writing A in its orthogonal rank-one sum decomposition...

■

To get a rough idea for why it's desirable to find the solution with smallest norm, let's return to the linear system

$$\begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ 0 \\ 6 \end{bmatrix}$$

from earlier. The solution set of this linear system consists of the vectors of the form

This solution set contains some vectors that are hideous, and some that are not so hideous:

The guarantee that the pseudoinverse finds the smallest-norm solution means that we do not have to worry about it returning "large and ugly" solutions like the first one above.

Geometrically, it means that the pseudoinverse finds the solution closest to the origin:

Not only does the pseudoinverse find the "best" solution when a solution exists, it even find the "best" non-solution when no solution exists!

This is strange to think about, but it makes sense if we again think in terms of norms and distances—if no solution to a linear system $A\mathbf{x} = \mathbf{b}$ exists, then it seems reasonable that the "next best thing" to a solution would be the vector that makes $A\mathbf{x}$ as close to $\mathbf{b}$ as possible. In other words, we want to find the vector $\mathbf{x}$ that...

The following theorem shows that choosing $\mathbf{x} = A^\dagger\mathbf{b}$ also solves this problem:

Theorem 10.2 — Linear Least Squares

Suppose $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$, $A \in \mathcal{M}_{m,n}(\mathbb{F})$, and $\mathbf{b} \in \mathbb{F}^m$. If $\mathbf{x} = A^\dagger\mathbf{b}$ then

We won't prove this theorem (see the textbook if you're curious), but it comes up a lot in statistics, since it can be used to fit data to a model. For example, suppose we had 4 data points:

and we want to find a line of best fit for those data points (i.e., a line with the property that the sum of squares of vertical distances between the points and the line is minimized). To find this line, we consider the "ideal" scenario—we try (and typically fail) to find a line that passes exactly through all n data points by setting up the corresponding linear system:

Since this linear system has 4 equations, but only 2 variables (m and b), we do not expect to find an exact solution, but we can find the closest thing to a solution by using the pseudoinverse:

This exact same method also works for finding the "plane of best fit" for data points $(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)$, and so on for higher-dimensional data as well. You can even do things like find quadratics of best fit, exponentials of best fit, or other weird functions of best fit.

By putting together all of the results of this section, we see that the pseudoinverse gives the "best solution" to a system of linear equations $A\mathbf{x} = \mathbf{b}$ in all cases:

The Operator Norm

We have seen one way of measuring the size of a matrix—the Frobenius norm. In practice, the Frobenius norm is actually not very useful (it's just used because it's easy to calculate), and the following norm is more commonly used instead:

Definition 10.2 — Operator Norm

Suppose $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$, and $A \in \mathcal{M}_{m,n}(\mathbb{F})$. Then the **operator norm** of A, denoted by $\|A\|$, is either of the following (equivalent) quantities:

The operator norm is the maximum amount by which a matrix can stretch a vector:

Before showing that $\|A\|$ really is the largest singular value of A, let's establish some of its more basic properties.

Theorem 10.3 — Submultiplicativity

Suppose $A \in \mathcal{M}_{m,n}$ and $B \in \mathcal{M}_{n,p}$. Then

Proof. Notice that a matrix $A \in \mathcal{M}_{m,n}$ cannot stretch any vector by more than a factor of $\|A\|$:

■

Theorem 10.4 — Unitary Invariance

Let $A \in \mathcal{M}_{m,n}$ and suppose $U \in \mathcal{M}_m$ and $V \in \mathcal{M}_n$ are unitary matrices. Then

Proof. We start by showing that every unitary matrix $U \in \mathcal{M}_m$ has $\|U\| = 1$:

■

As a side note, the previous two theorems both hold for the Frobenius norm as well (try to prove these facts on your own). That is,

By combining unitary invariance with the singular value decomposition, we almost immediately confirm our observation that the operator norm should equal the matrix's largest singular value, and we also get a new formula for the Frobenius norm:

Theorem 10.5 — Matrix Norms in Terms of Singular Values

Let $A \in \mathcal{M}_{m,n}$ have rank r and singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. Then

Proof. If we write A in its singular value decomposition $A = U\Sigma V^*$, then unitary invariance tells us that $\|A\| = \|\Sigma\|$ and $\|A\|_{\mathrm{F}} = \|\Sigma\|_{\mathrm{F}}$. Well,

■

Example. *Compute the operator and Frobenius norms of* $A = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 1 \\ 3 & 2 & 1 \end{bmatrix}$.

Low-Rank Approximation

As one final application of the singular value decomposition, we consider the problem of approximating a matrix by another matrix with small rank. One of the primary reasons why we would do this is that it allows us to compress data that is represented by a matrix, since a full $n \times n$ matrix requires us to store...

However, a rank-k matrix only requires us to store

Since $2kn$ is much smaller than n^2 when k is small, it is much less resource-intensive to store low-rank matrices than general matrices. Thus to compress data, instead of storing the exact matrix A that contains our data of interest, we can sometimes find a nearby matrix with small rank and store that instead.

To actually find a nearby low-rank matrix, we use the following theorem:

Theorem 10.6 — Eckart–Young–Mirsky

Suppose $\mathbb{F} = \mathbb{R}$ or $\mathbb{F} = \mathbb{C}$, and $A \in \mathcal{M}_{m,n}(\mathbb{F})$ has orthogonal rank-one sum decomposition

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$. Then the closest rank-$k$ matrix to A (i.e., the rank-k matrix B that minimizes $\|A - B\|$) is

In other words, the Eckart–Young–Mirsky theorem says that...

We skip the proof of the Eckart–Young–Mirsky theorem (see the textbook if you're curious), and instead jump right into a numerical example to illustrate its usage.

Example. *Find the closest rank-1 matrix to* $A = \begin{bmatrix} 1 & 2 & 3 \\ -1 & 0 & 1 \\ 3 & 2 & 1 \end{bmatrix}$.

It is also worth noting that the Eckart–Young–Mirsky theorem works for many other matrix norms as well (like the Frobenius norm)—not just the operator norm.

One of the most interesting applications of this theorem is that it lets us do (lossy) image compression. We can represent an image by...

Applying the Eckart–Mirsky–Young theorem to those matrices then lets us compress the image. For example, let's use the following image:

Let's use MATLAB to compress the image by truncating its matrices' singular value decompositions: