Lecture 19

Singular Value Decomposition
SVD & Least-Squares

Owen L. Lewis

Department of Mathematics and Statistics
University of New Mexico

Oct. 31, 2024
Happy Halloween!

Last time

We formulated Least-Squares problems.

Formed & solved the **normal equations**. Also illustrated some of the problems with normal equations (bad condition numbers).

Overdetermined Systems

In our example, we wanted to find a and b that solves

$$\begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Systems with more equations than unknowns are called **overdetermined** If A is an $m \times n$ matrix, then in general, an $m \times 1$ vector b may not lie in the column space of A. Hence $Ax = b$ may not have an exact solution.

Definition

The **residual** vector is

$$r = b - Ax.$$

The **least squares** solution is given by minimizing the square of the residual in the 2-norm.

Other approaches

- SVD - singular value decomposition
  - For $A \in \mathbb{R}^{m\times n}$, factor $A = USV^T$ where
    - U is an $m \times m$ orthogonal matrix
    - V is an $n \times n$ orthogonal matrix
    - S is an $m \times n$ diagonal matrix whose elements are the singular values.
- QR factorization.
  - For $A \in \mathbb{R}^{m\times n}$, factor $A = QR$ where
    - Q is an $m \times m$ orthogonal matrix
    - R is an $m \times n$ upper triangular matrix (since R is an $m \times n$ upper triangular matrix we can write $R = \begin{bmatrix} R' \\ 0 \end{bmatrix}$ where R is $n \times n$ upper triangular and 0 is the $(m-n) \times n$ matrix of zeros)

Eigen "stuff"

An $n \times n$ square matrix A has an "eigenvalue" λ if

$$A\vec{v} = \lambda\vec{v},$$

for some non-zero vector $\vec{v}$. We say that $\vec{v}$ is an "eigenvector" associated with the eigenvalue λ.

Notes

To transition

Unfortunately

Not all matrices (that we care about) have eigenvalues and eigenvectors!

What if A is $m \times n$, where $n \neq m$?

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad y = Ax = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

It is impossible to solve $y = \lambda x$ because y and x are not the same size!

Diagonalization

Theorem 1

If A is a “nice” $n \times n$ matrix, we can find a diagonal matrix D and an invertible matrix P such that

$$A = PDP^{-1}.$$

Theorem 2

If A is a <u>normal</u> matrix, we can find a diagonal matrix D and an orthogonal matrix V such that

$$A = VDV^T.$$

Here, V has the eigenvectors as its columns and D has the eigenvalues on the diagonal.

Theorem 2 is extremely nice, and there is “version” of it for matrices that aren’t even $n \times n$.

SVD: motivation

SVD uses in practice:

1. Search Technology: find closely related documents or images in a database
2. Clustering: aggregate documents or images into similar groups
3. Compression: efficient image storage
4. Principal axis: find the main axis of a solid (engineering/graphics)
5. Summaries: Given a textual document, ascertain the most representative tags
6. Graphs: partition graphs into subgraphs (graphics, analysis)

SVD: Singular Value Decomposition

SVD takes an $m \times n$ matrix A and factors it:

$$A = USV^T$$

where U $(m \times m)$ and V $(n \times n)$ are orthogonal and S $(m \times n)$ is diagonal.

Definition

U is orthogonal if $U^T U = UU^T = I$.

S is diagonal, with the diagonal entries made up of “singular values”:

$$\sigma_1 \geqslant \sigma_2 \geqslant \cdots \geqslant \sigma_r \geqslant \sigma_{r+1} = \cdots = \sigma_p = 0$$

Here, $r = rank(A)$ and $p = min(m, n)$.

Diagonalizing a matrix

We want to factorize A into U, S, and V^T. First step: find V. Consider

$$A = USV^T$$

and multiply by A^T

$$A^TA = (USV^T)^T(USV^T) = VS^TU^TUSV^T$$

Since U is orthogonal

$$A^TA = VS^2V^T$$

This is called a similarity transformation, i.e., A^TA and S^2 are similar.

Definition

Matrices A and B are similar if there is an invertible matrix Q such that

$$Q^{-1}BQ = A$$

Theorem

Similar matrices have the same eigenvalues.

Proof

Let v, λ be an eigenvector, eigenvalue pair for matrix B.

$$
\begin{aligned}
Bv &= \lambda v \\
Q^{-1}AQv &= \lambda v \\
AQv &= \lambda Qv \\
Aw &= \lambda w.
\end{aligned}
$$

Thus, $w = Qv$ is an eigenvector of A with the same eigenvalue λ.

So far...

Need $A = USV^T$

Look for V such that $A^TA = VS^2V^T$. Here S^2 is diagonal.

If A^TA and S^2 are similar, then they have the same eigenvalues. So the diagonal matrix S^2 is just the eigenvalues of A^TA and V is the matrix of eigenvectors. To see the latter, post-multiply both sides by V and use

$V^TV = I$,

$$A^TAV = VS^2$$

Looking at the i-th column, and see that you have an eigenvector

$$(A^TA)v_i = \sigma_i^2 v_i$$

Similarly...

Now consider

$$A = USV^T$$

and multiply by A^T from the right

$$AA^T = (USV^T)(USV^T)^T = USV^TVS^TU^T$$

Since V is orthogonal

$$AA^T = US^2U^T$$

Now U is the matrix of eigenvectors of AA^T.

In the end...

We get

$$A = \begin{bmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_m \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} \dots & v_1^T & \dots \\ \dots & \vdots & \dots \\ \dots & v_n^T & \dots \end{bmatrix}$$

Example

Decompose

$$A = \begin{bmatrix} 2 & -2 \\ 1 & 1 \end{bmatrix}$$

First construct $A^T A$:

$$A^T A = \begin{bmatrix} 2 & 1 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 5 & -3 \\ -3 & 5 \end{bmatrix}$$

Eigenvalues: $\lambda_1 = 8$ and $\lambda_2 = 2$. So

$$S^2 = \begin{bmatrix} 8 & 0 \\ 0 & 2 \end{bmatrix} \quad \Rightarrow \quad S = \begin{bmatrix} 2\sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix}$$

Example

Now find V^T and U. The columns of V^T are the eigenvectors of A^TA.

- $\lambda_1 = 8$: $(A^TA - \lambda_1 I)v_1 = 0$

$$\Rightarrow \begin{bmatrix} -3 & -3 \\ -3 & -3 \end{bmatrix} v_1 = 0 \quad \Rightarrow \quad \begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} v_1 = 0 \quad \Rightarrow \quad v_1 = \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}$$

- $\lambda_2 = 2$: $(A^TA - \lambda_2 I)v_2 = 0$

$$\Rightarrow \begin{bmatrix} 3 & -3 \\ -3 & 3 \end{bmatrix} v_2 = 0 \quad \Rightarrow \quad \begin{bmatrix} 1 & -1 \\ 0 & 0 \end{bmatrix} v_2 = 0 \quad \Rightarrow \quad v_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}$$

- Finally:

$$V = \begin{bmatrix} -\sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}$$

Example

Now find U. The columns of U are the eigenvectors of AA^T.

- $\lambda_1 = 8$: $(AA^T - \lambda_1 I)u_1 = 0$

$$\Rightarrow \begin{bmatrix} 0 & 0 \\ 0 & -6 \end{bmatrix} u_1 = 0 \quad \Rightarrow \quad \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} u_1 = 0 \quad \Rightarrow \quad u_1 = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

- $\lambda_2 = 2$: $(AA^T - \lambda_2 I)u_2 = 0$

$$\Rightarrow \begin{bmatrix} 6 & 0 \\ 0 & 0 \end{bmatrix} u_2 = 0 \quad \Rightarrow \quad \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} u_2 = 0 \quad \Rightarrow \quad u_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

- Finally:

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

- Together:

$$A = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2\sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} -\sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix}$$

SVD: who cares?

How can we actually *use* $A = USV^T$? We can use this to represent A with far fewer entries...

Notice what $A = USV^T$ looks like:

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T + 0 u_{r+1} v_{r+1}^T + \cdots + 0 u_p v_p^T$$

This is easily truncated to

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \cdots + \sigma_r u_r v_r^T$$

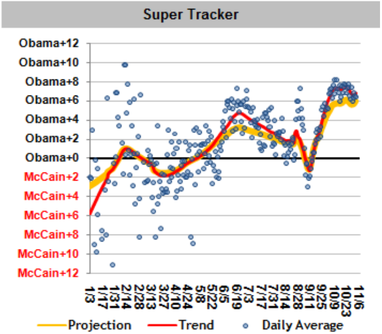
What are the savings?

- A takes $m \times n$ storage
- Using k terms of U and V takes $k(1 + m + n)$ storage

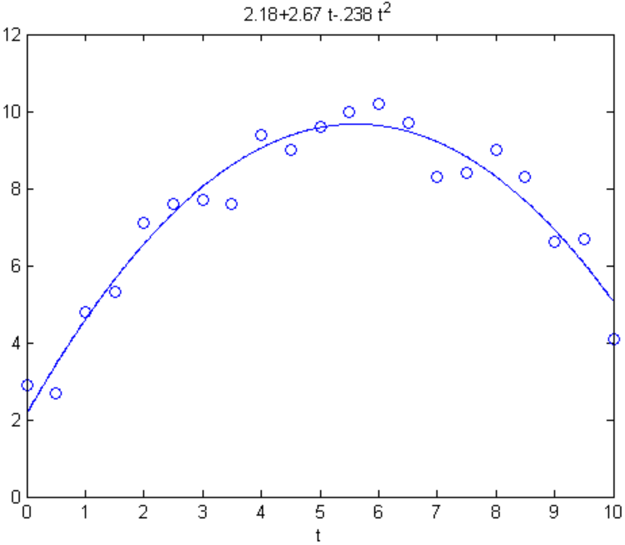
Polling data

Recall in interpolation we wanted to find a curve that went through *all* of the data points.

Suppose we are given the data $\{(x_1, y_1), ..., (x_n, y_n)\}$ and we want to find a curve that *best fits* the data.



Fitting curves



Fitting a line

Given n data points $\{(x_1, y_i), ..., (x_n, y_n)\}$ find a and b such that

$$y_i = ax_i - b \quad \forall i \in [1, n].$$

In matrix form, find a and b that solves

$$\begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_n & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Systems with more equations than unknowns are called **overdetermined**

Overdetermined Systems

If A is an $m \times n$ matrix, then in general, an $m \times 1$ vector b may not lie in the column space of A. Hence $Ax = b$ may not have an exact solution.

Definition

The **residual** vector is

$$r = b - Ax.$$

The **least squares** solution is given by minimizing the square of the residual in the 2-norm.

Using SVD for least squares

Recall that a singular value decomposition is given by

$$A = \begin{bmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_m \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} \dots & v_1^T & \dots \\ \dots & \vdots & \dots \\ \dots & v_n^T & \dots \end{bmatrix}$$

where σ_i are the singular values.

Using SVD for least squares

Assume that A has rank k (and hence k nonzero singular values σ_i) and recall that we want to minimize

$$||r||_2^2 = ||b - Ax||_2^2.$$

Substituting the SVD for A we find that

$$||r||_2^2 = ||b - Ax||_2^2 = ||b - USV^T x||_2^2$$

where U and V are orthogonal and S is diagonal with k nonzero singular values.

$$||b - USV^T x||_2^2 = ||U^T b - U^T USV^T x||_2^2 = ||U^T b - SV^T x||_2^2$$

Notes

Using SVD for least squares

Let $c = U^T b$ and $y = V^T x$ (and hence $x = Vy$) in $||U^T b - SV^T x||_2^2$. We now have

$$||r||_2^2 = ||c - Sy||_2^2$$

Since S has only k nonzero diagonal elements, we have

$$||r||_2^2 = \sum_{i=1}^{k} (c_i - \sigma_i y_i)^2 + \sum_{i=k+1}^{n} c_i^2$$

which is minimized when $y_i = \frac{c_i}{\sigma_i}$ for $1 \leqslant i \leqslant k$.

Using SVD for least squares

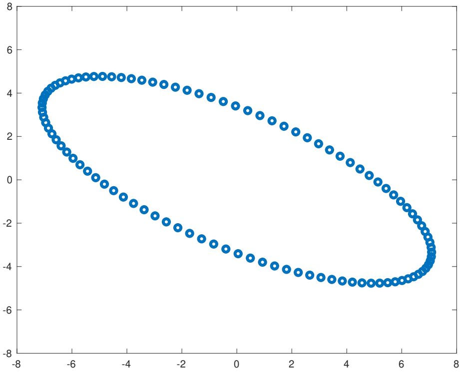
Theorem

Let A be an $m \times n$ matrix of rank r and let $A = USV^T$, the singular value decomposition. The least squares solution of the system $Ax = b$ is

$$x = \sum_{i=1}^{r} (\sigma_i^{-1} c_i) v_i$$

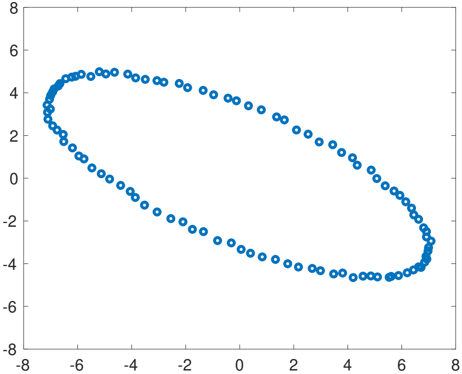
where $c_i = u_i^T b$.

Data Has shape!



How would we go about finding the “orientation” of this data?

Data Has shape!



What about this data?

Data Has shape!

Take all your data and put it in a giant array.

$$A = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_m & y_m \end{bmatrix}$$

Now calculate the SVD of this array....

Data Has shape!

$$A = USV^T$$

What do S and V tell us?

$$S = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} \vec{v}_1 & \vec{v}_2 \end{bmatrix}$$

Columns of V give us two perpendicular directions. These are the “natural” orientation of our data. The singular values σ give us the ”importance” of that direction.

This is sometimes called “principle component analysis”

SVD for Shape Analysis

Example

ellipse_SVD.m