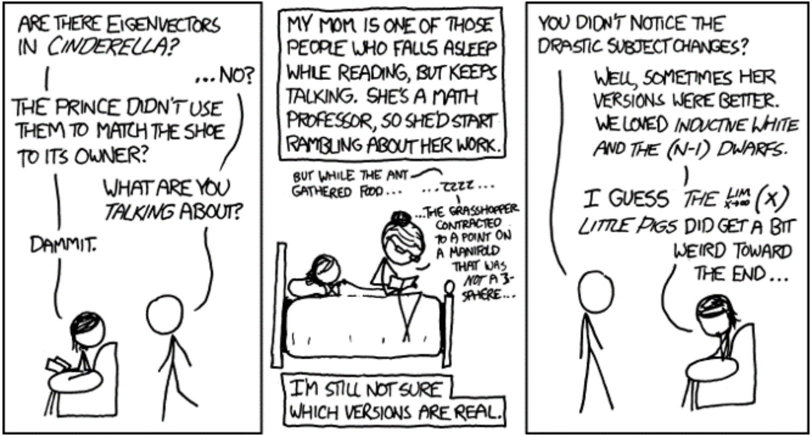
CSCI 4022 Spring 2021
Singular Value Decomposition


ARE THERE EIGENVECTORS IN *CINDERELLA*?
...NO?
THE PRINCE DIDN'T USE THEM TO MATCH THE SHOE TO ITS OWNER?
WHAT ARE YOU *TALKING* ABOUT?
DAMMIT.
MY MOM IS ONE OF THOSE PEOPLE WHO FALLS ASLEEP WHILE READING, BUT KEEPS TALKING. SHE'S A MATH PROFESSOR, SO SHE'D START RAMBLING ABOUT HER WORK.
BUT WHILE THE ANT GATHERED FOOD...
...ZZZZ...
...THE GRASSHOPPER CONTRACTED TO A POINT ON A MANIFOLD THAT WAS *NOT* A 3-SPHERE...
I'M STILL NOT SURE WHICH VERSIONS ARE REAL.
YOU DIDN'T NOTICE THE DRASTIC SUBJECT CHANGES?
WELL, SOMETIMES HER VERSIONS WERE BETTER. WE LOVED *INDUCTIVE WHITE AND THE (N-1) DWARFS.*
I GUESS *THE* $\lim_{x \to \infty}(x)$ *LITTLE PIGS* DID GET A BIT WEIRD TOWARD THE END...

Why Reduce Dimension

1. Discover hidden correlations/topics/concepts
2. Remove redundant/noisy features
3. Interpretation and visualization is easier and more intuitive in fewer dimensions
4. Easier to store, process and analyze data in fewer dimensions

Definition: $(\lambda, \boldsymbol{v})$ is an eigenpair of a matrix A if $A\boldsymbol{v} = \lambda\boldsymbol{v}$ and $v \neq 0$. So... $(A - \lambda I)\boldsymbol{v} = 0$ and since $|\boldsymbol{v}| = 1$, $|A - \lambda I| = 0$.

(pen-and-paper) Algorithm:Write down the determinant $|A - \lambda I|$ (a polynomial) and solving for its roots.

Then, we can set up and solve the linear system $A\boldsymbol{v} = \lambda\boldsymbol{v}$ (with the restriction that $|\boldsymbol{v}| = 1$) to find the associated eigenvector.

PCA; Example

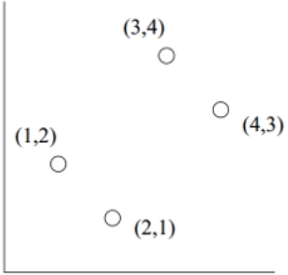
$$M = \begin{array}{cc} & \text{Features} \\ \text{observations} & \begin{bmatrix} \quad & \quad & \quad \end{bmatrix} \end{array} \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix}$$

We'll walk through a stylized and simplistic example to illustrate how PCA is done.

Example: Consider the set of data points (1,2), (2,1), (3,4), (4,3).

Step 1:

1. Construct M, and $M^T M$ (or $M M^T$, whichever is smaller)



Solution:

Theorem: For any real matrix A, $A^T A$ is symmetric.

PCA; Example

We'll walk through a stylized and simplistic example to illustrate how PCA is done.

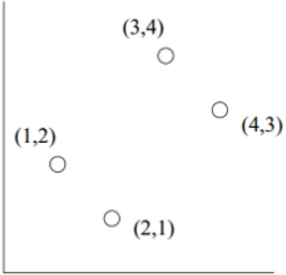
Example: Consider the set of data points (1,2), (2,1), (3,4), (4,3).

Step 1:

1. Construct M, and M^TM (or MM^T, whichever is smaller)

Solution:

$$M = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix}; \quad M^TM = \overset{2\times4}{\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{bmatrix}} \overset{4\times2}{\begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix}} = \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix}$$



Theorem: For any real matrix A, A^TA is symmetric.

PCA; Example

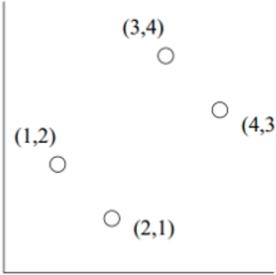
Example: Consider the set of data points (1,2), (2,1), (3,4), (4,3).

We have that $M^TM = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix}$

Step 2:

2. Find the eigenpairs for M^TM.

Solution:



PCA; Example

Example: Consider the set of data points (1,2), (2,1), (3,4), (4,3).

We have that $M^TM = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix} = \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix}$

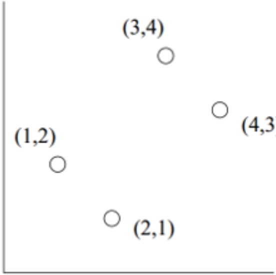
Step 2:

2. Find the eigenpairs for M^TM.

Solution:

$$|M^TM - \lambda I| = | \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix} | = (30-\lambda)^2 - 28^2$$

which has eigenvalues of $\lambda_1 = 58$, $\lambda_2 = 2$.



PCA; Example

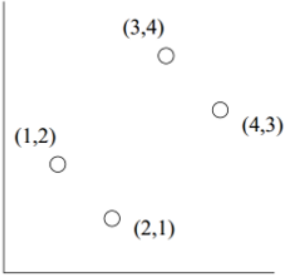
Example: Consider the set of data points (1,2), (2,1), (3,4), (4,3).

We have that $\lambda_1 = 58$, $\lambda_2 = 2$

Step 2b:

2b. Find the eigenvectors for $A^T A$.

Solution:



PCA; Example

Example: Consider the set of data points (1,2), (2,1), (3,4), (4,3).

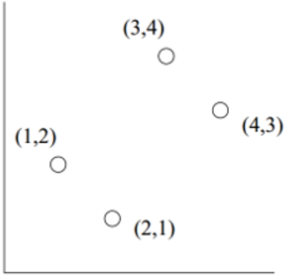
We have that $\lambda_1 = 58$, $\lambda_2 = 2$

Step 2b:

2b. Find the eigenvectors for $A^T A$.

Solution: For $\lambda_1 = 58$:

$$M^T M v_1 = \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \overset{\heartsuit}{=} 58 \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$



$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

PCA; Example

Example: Consider the set of data points (1,2), (2,1), (3,4), (4,3).

We have that $\lambda_1 = 58$, $\lambda_2 = 2$

Step 2b:

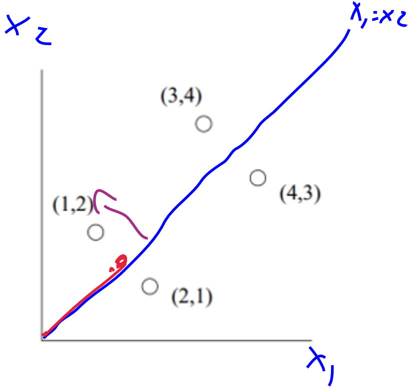
2b. Find the eigenvectors for A^TA.

Solution: For $\lambda_1 = 58$:

$$M^TMv_1 = \begin{bmatrix} 30 & 28 \\ 28 & 30 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \overset{\heartsuit}{=} 58 \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

so $30v_1 + 28v_2 = 58v_1$, which has solution of $\boxed{v_1 = v_2}$ and normalizes to $\begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$

Similarly, $v_2 = \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$



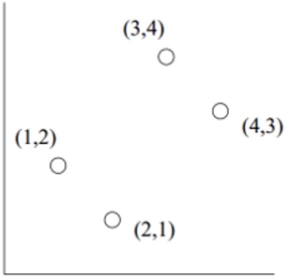
The Rotation

So we have the eigenpairs of $M^T M$.

Step 3:

3. Step 3. Construct E, the matrix whose *columns* are made up of the eigenvectors.

Solution:



The Rotation

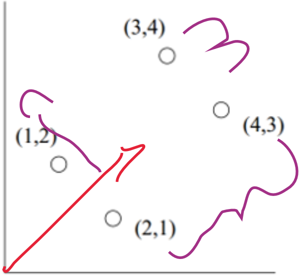
So we have the eigenpairs of $M^T M$.

Step 3:

3. Step 3. Construct E, the matrix whose *columns* are made up of the eigenvectors.

Solution:

$$E = \begin{bmatrix} \uparrow & \uparrow \\ v_1 & v_2 \\ \downarrow & \downarrow \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$



The Rotation

So we have the eigenpairs of M^TM.

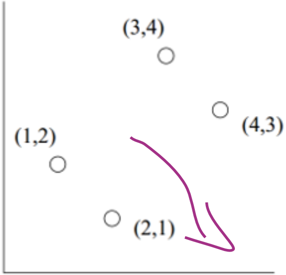
Step 3:

3. Step 3. Construct E, the matrix whose *columns* are made up of the eigenvectors.

Solution:

$$E = \begin{bmatrix} \uparrow & \uparrow \\ v_1 & v_2 \\ \downarrow & \downarrow \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$

$\theta : \pi/4$



Definition: Any matrix - such as E - whose columns are made up of **orthonormal vectors** represents a **rotation** or **reflection** in a Euclidean space.

Example: $\boxed{T = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}}$ is the 2-D matrix that rotates clockwise by θ

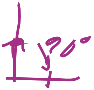
Rotations

How does that work?

Example: Example: Consider the row vector $x = [0, 1]$ in $\mathbb{R}^2$. What is the matrix T that will rotate it clockwise by 90°? Perform this transformation xT to find the resulting rotated vector.

Solution:

Rotations



How does that work?

Example: Example: Consider the row vector $x = [0, 1]$ in $\mathbb{R}^2$. What is the matrix T that will rotate it clockwise by $90°$? Perform this transformation xT to find the resulting rotated vector.

Solution:

$$T = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \boxed{\begin{bmatrix} \cos\pi/2 & -\sin\pi/2 \\ \sin\pi/2 & \cos\pi/2 \end{bmatrix}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

so $xT = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix}$ And [1, 0] is indeed what we expect if we were to rotate $[0, 1]$ clockwise by $90°$. Hooray!

Example, cont'd: Our rotation matrix is $E = \begin{bmatrix} \uparrow & \uparrow \\ v_1 & v_2 \\ \downarrow & \downarrow \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$

Rotations

How does that work?

Example: Example: Consider the row vector $x = [0, 1]$ in $\mathbb{R}^2$. What is the matrix T that will rotate it clockwise by $90°$? Perform this transformation xT to find the resulting rotated vector.

Solution:

$$T = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \cos\pi/2 & -\sin\pi/2 \\ \sin\pi/2 & \cos\pi/2 \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

so $xT = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix}$ And [1, 0] is indeed what we expect if we were to rotate $[0, 1]$ clockwise by $90°$. Hooray!

Example, cont'd: Our rotation matrix is $E = \begin{bmatrix} \uparrow & \uparrow \\ v_1 & v_2 \\ \downarrow & \downarrow \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$, so expect it will rotate our original data points by $45°$ (since $\cos^{-1}(1/\sqrt{2}) = \pi/4$.)

Rotations and PCA

So we have the *rotation* represented by $E = \begin{bmatrix} \uparrow & \uparrow \\ v_1 & v_2 \\ \downarrow & \downarrow \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$

Step 4:

4. Rotate the original data points M by E. This puts them into a new frame of reference:

Solution:

Rotations and PCA

So we have the *rotation* represented by $E = \begin{bmatrix} \uparrow & \uparrow \\ v_1 & v_2 \\ \downarrow & \downarrow \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$

Step 4:

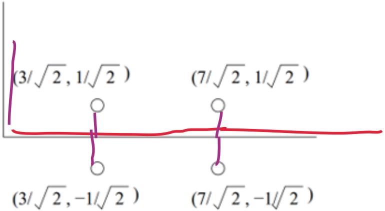
4. Rotate the original data points M by E. This puts them into a new frame of reference:

Solution:

$$ME = \overset{4\times2}{\begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix}} \overset{2\times2}{\begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}} = \overset{4\times2}{\begin{bmatrix} 3/\sqrt{2} & 1/\sqrt{2} \\ 3/\sqrt{2} & -1/\sqrt{2} \\ 7/\sqrt{2} & 1/\sqrt{2} \\ 7/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}}$$

most relevant

remainders



Rotations and PCA

So we have the *rotation* represented by $E = \begin{bmatrix} \uparrow & \uparrow \\ v_1 & v_2 \\ \downarrow & \downarrow \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$

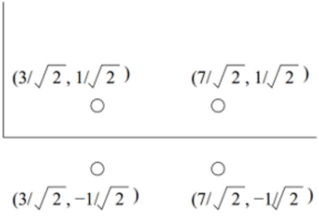
Step 4:

4. Rotate the original data points M by E. This puts them into a new frame of reference:

$\frac{3}{\sqrt{2}}\begin{bmatrix}1\\1\end{bmatrix} + \frac{1}{\sqrt{2}}\begin{bmatrix}-1\\1\end{bmatrix}$

Solution:

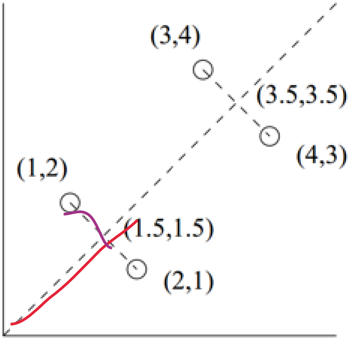
$$ME = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 3/\sqrt{2} & 1/\sqrt{2} \\ 3/\sqrt{2} & -1/\sqrt{2} \\ 7/\sqrt{2} & 1/\sqrt{2} \\ 7/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$



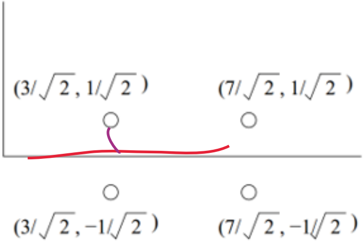
Upshot: The rows of ME give the coordinates in theeigenvector basis (along the principal components) of the corresponding rows (data points) in M.

Rotations and PCA

$$ME = \begin{bmatrix} 3/\sqrt{2} & 1/\sqrt{2} \\ 3/\sqrt{2} & -1/\sqrt{2} \\ 7/\sqrt{2} & 1/\sqrt{2} \\ 7/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$ *rotates* M into a new frame of reference.



Rotates to



Sanity Check: Does the geometry work out?

Rotations and PCA

$$ME = \begin{bmatrix} 3/\sqrt{2} & 1/\sqrt{2} \\ 3/\sqrt{2} & -1/\sqrt{2} \\ 7/\sqrt{2} & 1/\sqrt{2} \\ 7/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$$

The rows of ME give the coordinates in the **eigenvector basis** (along the principal components) of the corresponding rows (data points) in M.

In other words, if we took $3/\sqrt{2}x_1 + 1/\sqrt{2}x_2$, we'd have a new location for datum $\#1$.

Interpretation: The components in ME give the distance to each data point along the eigenvectors.

...and reduce

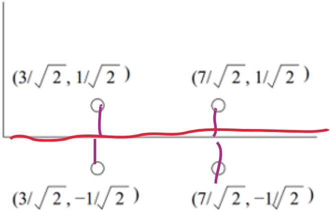
So we have the *projection* represented by ME... but let's not use all of it!

Step 5:

5. To get a reduced dimension representation of our data in M, can take only the eigenvectors associated with the k largest eigenvalues of M^TM (or MM^T).

Put them in a *narrower* matrix, E_k.

Then ME_k is a reduced-dimension representation of M,in the directions of the *most important* eigenvectors.



Back to the running **example**: There are only $D = 2$ dimensions, so the only reducing we can do is to $d = 1$. So take the principal eigenvector and reduce:

$$ME_1 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 4 \\ 4 & 3 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 3/\sqrt{2} \\ 3/\sqrt{2} \\ 7/\sqrt{2} \\ 7/\sqrt{2} \end{bmatrix}$$

... is an *approximation* of M with less columns! How close is it?

PCA: All told

1. Construct M, and $M^T M$ (or MM^T, whichever is smaller)
2. Find the eigenpairs for that matrix.
3. Construct E, the matrix whose *columns* are made up of the eigenvectors.
4. Rotate the original data points M by E.
5. To get a reduced dimension representation of our data in M, can take only the eigenvectors associated with the k largest eigenvalues of $M^T M$ (or MM^T).

Then ME_k is a reduced-dimension representation of M,in the directions of the *most important* eigenvectors.

Why bother?

There are two algorithms we've seen that PCA plays well with:

1. Dimension reduction that encourages orthonormal axes for the principal components helps with multiple linear regression (and logistic variants).

2. Clustering is more numerically stable with lower dimension... it also visualizes better! See also: multidimensional scaling.

One thing to note: our "approximation" of M was a truncated *rotation* $ME \to ME_1$. It might make sense to rotate that approximation *back* into the original units of M, since ME_1 has been rotated into the frame of references of that original largest component. That way, we can actually directly compare M to it's lower-dimensional representation! (We do this ~~next~~ time!)

Why bother?

There are two algorithms we've seen that PCA plays well with:

1. Dimension reduction that encourages orthonormal axes for the principal components helps with multiple linear regression (and logistic variants).
   Why?: if feature/columns are similar/dependent, then MLR inference fails due to $\hat{\beta} \propto \boxed{(X^T X)^{-1}}$, which is singular if columns of X are identical to one another.
2. Clustering is more numerically stable with lower dimension... it also visualizes better! See also: multidimensional scaling.
   Why?: in too many dimensions, distances tend to all look similar.

One thing to note: our "approximation" of M was a truncated *rotation* $ME \to ME_1$. It might make sense to rotate that approximation *back* into the original units of M, since ME_1 has been rotated into the frame of references of that original largest component. That way, we can actually directly compare M to it's lower-dimensional representation! (We do this next time!)

SVD

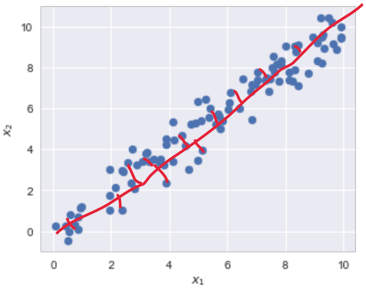
So the point of principal component analysis is to *reduce* a data matrix M into the effects of its **most important** eigenvectors. This is a subproblem of a larger problem: what is the *most important axis* or direction in a matrix. Clearly the principal eigenvector is a (crucial!) piece of that direction, but we might phrase our question in such a way that the direction requires some combination of *all* of the eigenvectors.

Goal: Discover the axis of the data

Example: The direction of $[1, 1]$ is most "important"

Process: Describe each point by how far from 0 it is along this direction

Singular value decomposition (SVD):



Gives the "best" (minimum sum of squared errors) axis on which to *project* our data
–**Best** means minimizes reconstruction error

Singular Value Decomposition (SVD)

Big idea: Suppose you have an $m \times n$ matrix A, with $rank(A) = r$, such as:

- m documents, n terms to compute similarities, make recommendations, do science
- m users, n movies
- Generally: m data points, n entries per data point

Like PCA, we want to decompose data into the most important axes, or concepts...
The Decomposition: $A = U\Sigma V^T$ with sizes: $A_{m\times n} = \boxed{U_{m\times r}}\boxed{\Sigma_{r\times r}}\boxed{V^T_{n\times r}}$.
Where each piece is:

rotation

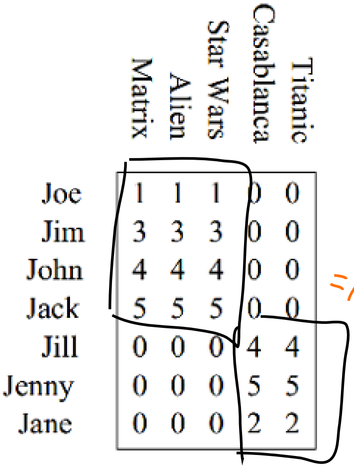
1. $\boxed{U_{m\times r}}$ left singular vectors (orthonormal); e.g, m documents, r concepts; orthonormal
2. $\Sigma_{r\times r}$ diagonal matrix of singular values, the strength of each concept, in decreasing order
3. $\boxed{V_{n\times r}}$ right singular vectors (orthonormal); n terms, r concepts; one vector per concept

SVD Example, Rank 2

Suppose the matrix M represents ratings of 5 movies by a set of 7 users.

We can find the SVD for M as:

$M = {}_{7}[\quad]^{2}\ {}_{2}[\quad]^{2}\ {}_{2}[\quad]^{5}$



	Matrix	Alien	Star Wars	Casablanca	Titanic
Joe	1	1	1	0	0
Jim	3	3	3	0	0
John	4	4	4	0	0
Jack	5	5	5	0	0
Jill	0	0	0	4	4
Jenny	0	0	0	5	5
Jane	0	0	0	2	2

$= M$

SVD Example, Rank 2

Suppose the matrix M represents ratings of 5 movies by a set of 7 users.

We can find the SVD for M as:

$$\begin{array}{c|ccccc|}
 & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\
\hline
\text{Joe} & \boxed{1} & 1 & 1 & 0 & 0 \\
\text{Jim} & 3 & 3 & 3 & 0 & 0 \\
\text{John} & 4 & 4 & 4 & 0 & 0 \\
\text{Jack} & 5 & 5 & 5 & 0 & 0 \\
\text{Jill} & 0 & 0 & 0 & 4 & 4 \\
\text{Jenny} & 0 & 0 & 0 & 5 & 5 \\
\text{Jane} & 0 & 0 & 0 & 2 & 2 \\
\hline
\end{array}
=
\underbrace{\begin{bmatrix} \cancel{.14} & \cancel{0} \\ .42 & 0 \\ .56 & 0 \\ .70 & 0 \\ 0 & .60 \\ 0 & .75 \\ 0 & .30 \end{bmatrix}}_{U}
\underbrace{\begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix}}_{\Sigma}
\underbrace{\begin{bmatrix} .58 & .58 & .58 & 0 & 0 \\ 0 & 0 & 0 & .71 & .71 \end{bmatrix}}_{V^T}$$

SVD Example, Rank 3

Suppose the matrix M represents ratings of 5 movies by a set of 7 users.

What if Jill and Jane watch *Alien* and rate it?

	Matrix	Alien	Star Wars	Casablanca	Titanic
Joe	1	1	1	0	0
Jim	3	3	3	0	0
John	4	4	4	0	0
Jack	5	5	5	0	0
Jill	0	2	0	4	4
Jenny	0	0	0	5	5
Jane	0	1	0	2	2

SVD Example, Rank 3

Suppose the matrix M represents ratings of 5 movies by a set of 7 users.

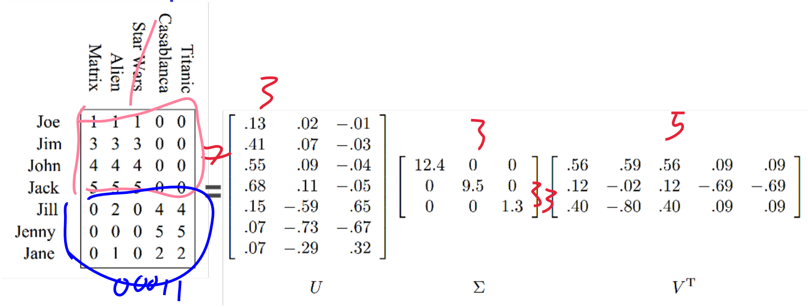
What if Jill and Jane watch *Alien* and rate it?

$$\begin{array}{c|ccccc|} & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ \hline \text{Joe} & 1 & 1 & 1 & 0 & 0 \\ \text{Jim} & 3 & 3 & 3 & 0 & 0 \\ \text{John} & 4 & 4 & 4 & 0 & 0 \\ \text{Jack} & 5 & 5 & 5 & 0 & 0 \\ \text{Jill} & 0 & 2 & 0 & 4 & 4 \\ \text{Jenny} & 0 & 0 & 0 & 5 & 5 \\ \text{Jane} & 0 & 1 & 0 & 2 & 2 \\ \hline \end{array} = \underset{U}{\begin{bmatrix} .13 & .02 & -.01 \\ .41 & .07 & -.03 \\ .55 & .09 & -.04 \\ .68 & .11 & -.05 \\ .15 & -.59 & .65 \\ .07 & -.73 & -.67 \\ .07 & -.29 & .32 \end{bmatrix}} \underset{\Sigma}{\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & \underset{\text{small}}{1.3} \end{bmatrix}} \underset{V^{\mathrm{T}}}{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \\ .40 & -.80 & .40 & .09 & .09 \end{bmatrix}}$$

Result: The matrix is now rank 3, so it has 3 *singular values*

1 11 00

SVD Example, Rank 3

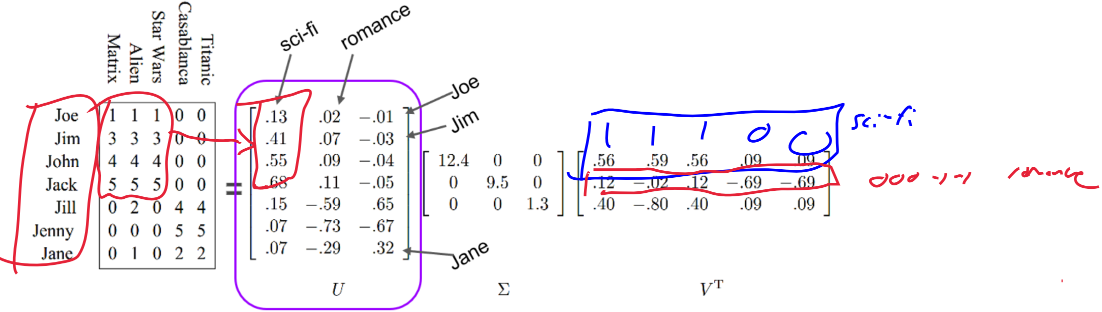


$$\begin{array}{l|ccccc} & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ \hline \text{Joe} & 1 & 1 & 1 & 0 & 0 \\ \text{Jim} & 3 & 3 & 3 & 0 & 0 \\ \text{John} & 4 & 4 & 4 & 0 & 0 \\ \text{Jack} & 5 & 5 & 5 & 0 & 0 \\ \text{Jill} & 0 & 2 & 0 & 4 & 4 \\ \text{Jenny} & 0 & 0 & 0 & 5 & 5 \\ \text{Jane} & 0 & 1 & 0 & 2 & 2 \\ & 0 & 0 & 0 & 1 & 1 \end{array} \overset{7}{=} \underbrace{\overset{3}{\begin{bmatrix} .13 & .02 & -.01 \\ .41 & .07 & -.03 \\ .55 & .09 & -.04 \\ .68 & .11 & -.05 \\ .15 & -.59 & .65 \\ .07 & -.73 & -.67 \\ .07 & -.29 & .32 \end{bmatrix}}}_{U} \underbrace{\overset{3}{\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}}}_{\Sigma} \; 3\,3 \; \underbrace{\overset{5}{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \\ .40 & -.80 & .40 & .09 & .09 \end{bmatrix}}}_{V^{\mathrm{T}}}$$

An SVD yields r *singular values* σ_k which represent the relative **strength of concepts**.

1. $U_{m\times r}$ maps users to concepts
2. $\Sigma_{r\times r}$ shows strength/importance of concepts
3. $V_{n\times r}$ maps movies to concepts

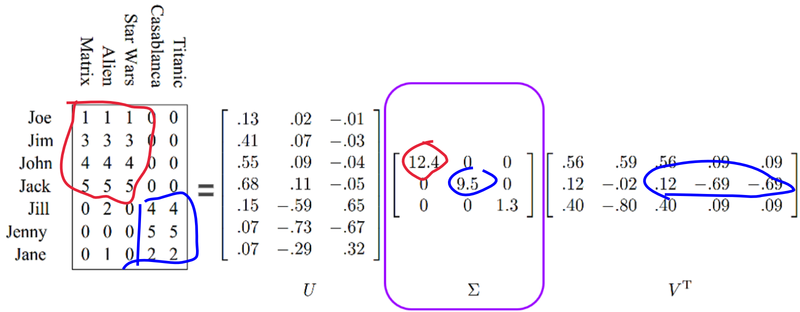
Intuition: what are the "concepts" in our toy example problem?



$$
\begin{array}{r|ccccc|}
 & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\
\hline
\text{Joe} & 1 & 1 & 1 & 0 & 0 \\
\text{Jim} & 3 & 3 & 3 & 0 & 0 \\
\text{John} & 4 & 4 & 4 & 0 & 0 \\
\text{Jack} & 5 & 5 & 5 & 0 & 0 \\
\text{Jill} & 0 & 2 & 0 & 4 & 4 \\
\text{Jenny} & 0 & 0 & 0 & 5 & 5 \\
\text{Jane} & 0 & 1 & 0 & 2 & 2 \\
\hline
\end{array}
=
\underbrace{\begin{bmatrix}
.13 & .02 & -.01 \\
.41 & .07 & -.03 \\
.55 & .09 & -.04 \\
.68 & .11 & -.05 \\
.15 & -.59 & .65 \\
.07 & -.73 & -.67 \\
.07 & -.29 & .32
\end{bmatrix}}_{U}
\underbrace{\begin{bmatrix}
12.4 & 0 & 0 \\
0 & 9.5 & 0 \\
0 & 0 & 1.3
\end{bmatrix}}_{\Sigma}
\underbrace{\begin{bmatrix}
.56 & .59 & .56 & .09 & .09 \\
.12 & -.02 & .12 & -.69 & -.69 \\
.40 & -.80 & .40 & .09 & .09
\end{bmatrix}}_{V^{\mathrm{T}}}
$$

An SVD yields r *singular values* σ_k which represent the relative **strength of concepts**.

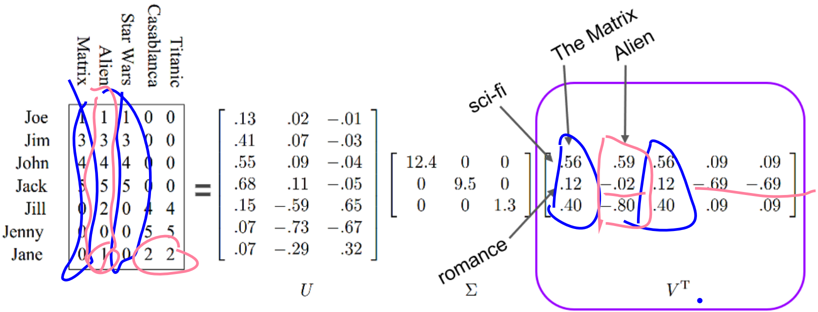
1. $U_{m\times r}$ maps users to concepts
2. $\Sigma_{r\times r}$ shows strength/importance of concepts
3. $V_{n\times r}$ maps movies to concepts



$$\begin{array}{c|ccccc} & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ \hline \text{Joe} & 1 & 1 & 1 & 0 & 0 \\ \text{Jim} & 3 & 3 & 3 & 0 & 0 \\ \text{John} & 4 & 4 & 4 & 0 & 0 \\ \text{Jack} & 5 & 5 & 5 & 0 & 0 \\ \text{Jill} & 0 & 2 & 0 & 4 & 4 \\ \text{Jenny} & 0 & 0 & 0 & 5 & 5 \\ \text{Jane} & 0 & 1 & 0 & 2 & 2 \end{array} = \underbrace{\begin{bmatrix} .13 & .02 & -.01 \\ .41 & .07 & -.03 \\ .55 & .09 & -.04 \\ .68 & .11 & -.05 \\ .15 & -.59 & .65 \\ .07 & -.73 & -.67 \\ .07 & -.29 & .32 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \\ .40 & -.80 & .40 & .09 & .09 \end{bmatrix}}_{V^{\mathrm{T}}}$$

An SVD yields r *singular values* σ_k which represent the relative **strength of concepts**.

1. $U_{m\times r}$ maps users to concepts
2. $\boxed{\Sigma_{r\times r} \text{ shows strength/importance of concepts}}$
3. $V_{n\times r}$ maps movies to concepts

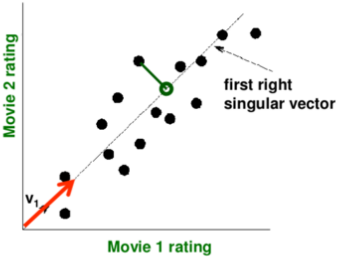


$$\begin{array}{c|ccccc} & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ \hline \text{Joe} & 1 & 1 & 1 & 0 & 0 \\ \text{Jim} & 3 & 3 & 3 & 0 & 0 \\ \text{John} & 4 & 4 & 4 & 0 & 0 \\ \text{Jack} & 5 & 5 & 5 & 0 & 0 \\ \text{Jill} & 0 & 2 & 0 & 4 & 4 \\ \text{Jenny} & 0 & 0 & 0 & 5 & 5 \\ \text{Jane} & 0 & 1 & 0 & 2 & 2 \end{array} = \underbrace{\begin{bmatrix} .13 & .02 & -.01 \\ .41 & .07 & -.03 \\ .55 & .09 & -.04 \\ .68 & .11 & -.05 \\ .15 & -.59 & .65 \\ .07 & -.73 & -.67 \\ .07 & -.29 & .32 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \\ .40 & -.80 & .40 & .09 & .09 \end{bmatrix}}_{V^{\mathrm{T}}}$$

An SVD yields r *singular values* σ_k which represent the relative **strength of concepts**.

1. $U_{m \times r}$ maps users to concepts
2. $\Sigma_{r \times r}$ shows strength/importance of concepts
3. $\boxed{V_{n \times r} \text{ maps movies to concepts}}$

Dimension Reduction with SVD



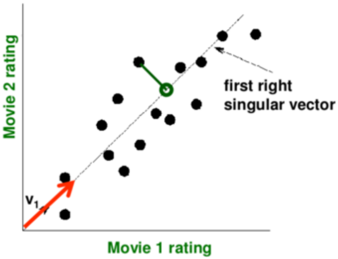
1-D, 2-D, n-D

Just like the related PCA, SVD can be used to reduce dimension.

SVD gives the “best” (minimum sum of squared errors) axis to project our data – minimizes reconstruction error

1. First right singular vector = axis on which to project
2. First singular value gives the spread (variance)
3. Connection to PCA: most important axes, and spread along those axes

Dimension Reduction with SVD



$$U\Sigma = \begin{bmatrix} 1.612 & 0.19 & -0.013 \\ 5.084 & 0.665 & -0.039 \\ 6.82 & 0.855 & -0.52 \\ 8.432 & 1.045 & -0.065 \\ 1.86 & -5.605 & 0.845 \\ 0.868 & -6.935 & -0.871 \\ 0.868 & -2.755 & 0.416 \end{bmatrix}$$

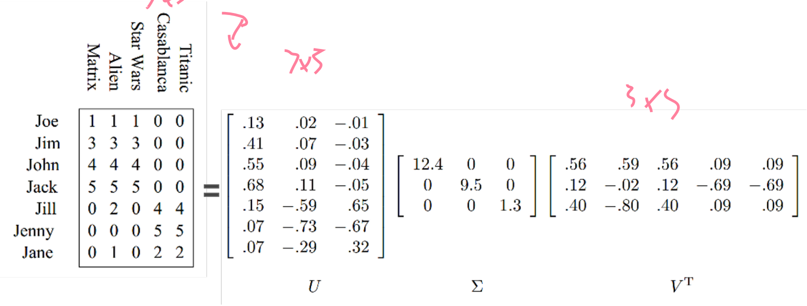
What defines the new locations of the data values?

1. The same as in PCA! We're taking a matrix of orthonormal column vectors and using it to *rotate* the data in Euclidean space.
2. The projection axis coordinates are the *concept space locations* of the movies $U\Sigma$.

Dimension Reduction with SVD

... but that's still 3D, and includes all the information from M. How do we actually *reduce* dimension? 7x5

?



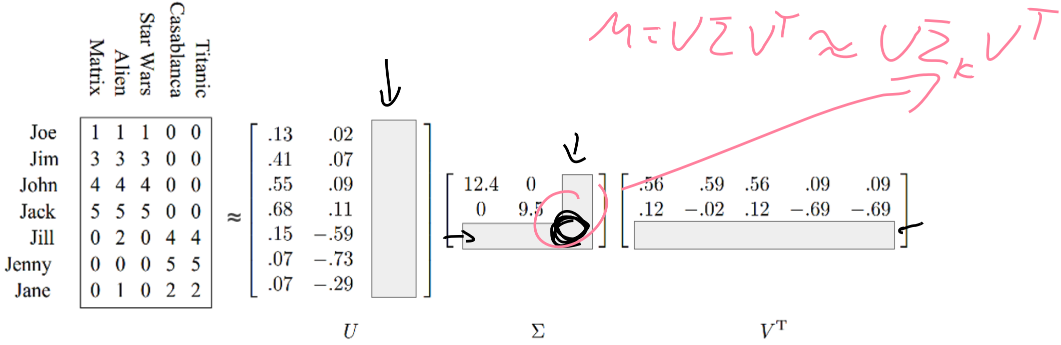
$$\begin{array}{c|ccccc} & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ \hline \text{Joe} & 1 & 1 & 1 & 0 & 0 \\ \text{Jim} & 3 & 3 & 3 & 0 & 0 \\ \text{John} & 4 & 4 & 4 & 0 & 0 \\ \text{Jack} & 5 & 5 & 5 & 0 & 0 \\ \text{Jill} & 0 & 2 & 0 & 4 & 4 \\ \text{Jenny} & 0 & 0 & 0 & 5 & 5 \\ \text{Jane} & 0 & 1 & 0 & 2 & 2 \end{array} = \underbrace{\overset{7\times 5}{\begin{bmatrix} .13 & .02 & -.01 \\ .41 & .07 & -.03 \\ .55 & .09 & -.04 \\ .68 & .11 & -.05 \\ .15 & -.59 & .65 \\ .07 & -.73 & -.67 \\ .07 & -.29 & .32 \end{bmatrix}}}_{U} \underbrace{\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}}_{\Sigma} \underbrace{\overset{3\times 5}{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \\ .40 & -.80 & .40 & .09 & .09 \end{bmatrix}}}_{V^{\mathrm{T}}}$$

Answer: we *discard* smaller singular values by setting them equal to zero.

Dimension Reduction with SVD

How do we actually *reduce* dimension?

1. **Answer:** we *discard* smaller singular values by setting them equal to zero.
2. **Result:** corresponding *columns* of U and *rows* of V^T are no longer used
3. Our modified system is a **low-rank** approximation of M. $M' = U\Sigma_{k}V^T \approx M$.



$$M = U\Sigma V^T \approx U\Sigma_k V^T$$

$$\begin{array}{c|ccccc|}
 & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\
\text{Joe} & 1 & 1 & 1 & 0 & 0 \\
\text{Jim} & 3 & 3 & 3 & 0 & 0 \\
\text{John} & 4 & 4 & 4 & 0 & 0 \\
\text{Jack} & 5 & 5 & 5 & 0 & 0 \\
\text{Jill} & 0 & 2 & 0 & 4 & 4 \\
\text{Jenny} & 0 & 0 & 0 & 5 & 5 \\
\text{Jane} & 0 & 1 & 0 & 2 & 2
\end{array}
\approx
\underset{U}{\begin{bmatrix} .13 & .02 \\ .41 & .07 \\ .55 & .09 \\ .68 & .11 \\ .15 & -.59 \\ .07 & -.73 \\ .07 & -.29 \end{bmatrix}}
\underset{\Sigma}{\begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix}}
\underset{V^T}{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \end{bmatrix}}$$

SVD Dimension Reduction

How similar is our approximation (M') to the original matrix (M)?

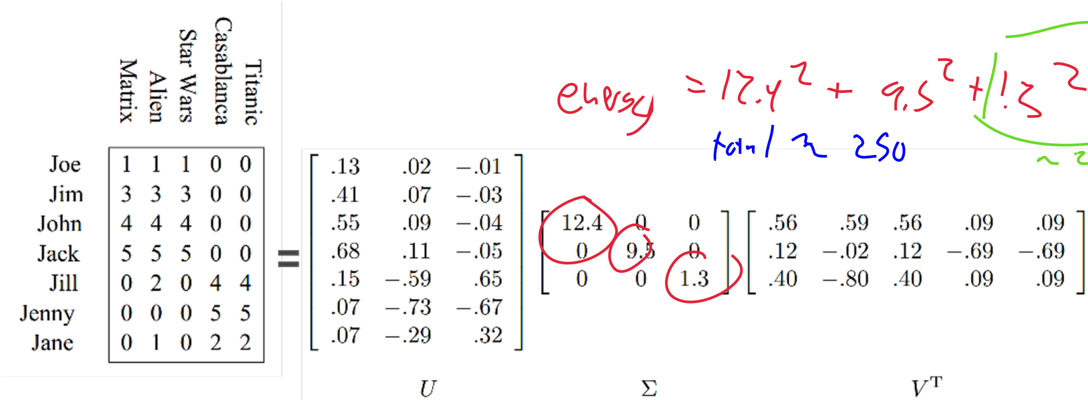
- Check using the Frobenius norm: $||A||_F = \sqrt{\sum_{i,j} A_{i,j}^2}$
- The actual difference between M and M' is $||M - M'||_F = \sqrt{\sum_{i,j} \left(M_{i,j} - M'_{i,j}\right)^2}$

$$\begin{array}{c|ccccc|}
 & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\
\hline
\text{Joe} & 1 & 1 & 1 & 0 & 0 \\
\text{Jim} & 3 & 3 & 3 & 0 & 0 \\
\text{John} & 4 & 4 & 4 & 0 & 0 \\
\text{Jack} & 5 & 5 & 5 & 0 & 0 \\
\text{Jill} & 0 & 2 & 0 & 4 & 4 \\
\text{Jenny} & 0 & 0 & 0 & 5 & 5 \\
\text{Jane} & 0 & 1 & 0 & 2 & 2 \\
\hline
\end{array}
\qquad
\begin{bmatrix}
0.93 & 0.95 & 0.93 & .014 & .014 \\
2.93 & 2.99 & 2.93 & .000 & .000 \\
3.92 & 4.01 & 3.92 & .026 & .026 \\
4.84 & 4.96 & 4.84 & .040 & .040 \\
0.37 & 1.21 & 0.37 & 4.04 & 4.04 \\
0.35 & 0.65 & 0.35 & 4.87 & 4.87 \\
0.16 & 0.57 & 0.16 & 1.98 & 1.98
\end{bmatrix}$$

SVD Example

How do we decide how many σ 's to keep?

- Define the *energy* as $\Sigma_i \sigma_i^2$
- General rule: keep enough singular values to account for 80-90% of the energy



$$\begin{array}{c|ccccc|}
 & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\
\hline
\text{Joe} & 1 & 1 & 1 & 0 & 0 \\
\text{Jim} & 3 & 3 & 3 & 0 & 0 \\
\text{John} & 4 & 4 & 4 & 0 & 0 \\
\text{Jack} & 5 & 5 & 5 & 0 & 0 \\
\text{Jill} & 0 & 2 & 0 & 4 & 4 \\
\text{Jenny} & 0 & 0 & 0 & 5 & 5 \\
\text{Jane} & 0 & 1 & 0 & 2 & 2 \\
\hline
\end{array}
=
\underset{U}{\begin{bmatrix}
.13 & .02 & -.01 \\
.41 & .07 & -.03 \\
.55 & .09 & -.04 \\
.68 & .11 & -.05 \\
.15 & -.59 & .65 \\
.07 & -.73 & -.67 \\
.07 & -.29 & .32
\end{bmatrix}}
\underset{\Sigma}{\begin{bmatrix}
12.4 & 0 & 0 \\
0 & 9.5 & 0 \\
0 & 0 & 1.3
\end{bmatrix}}
\underset{V^{\mathrm{T}}}{\begin{bmatrix}
.56 & .59 & .56 & .09 & .09 \\
.12 & -.02 & .12 & -.69 & -.69 \\
.40 & -.80 & .40 & .09 & .09
\end{bmatrix}}$$

$\text{energy} = 12.4^2 + 9.5^2 + 1.3^2$

total ≈ 250

$\sim 2/250$

SVD utility: concept-space

Suppose we have the previous simpler example below, and a new user, Megan, comes along. She has only watched The Matrix, and rated it a 4.

So, Megan's movie vector is $m = [4, 0, 0, 0, 0]$ We can map Megan into concept-space by multiplying mV (V maps movies → concepts). We find that $mv =$

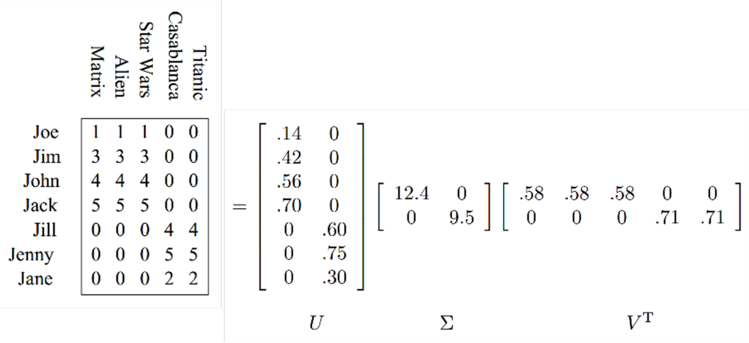
$$\begin{bmatrix} 4 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} .58 & 0 \\ .58 & 0 \\ .58 & 0 \\ 0 & .71 \\ 0 & .71 \end{bmatrix} = \begin{bmatrix} 2.32 & 0 \end{bmatrix}$$

, so we can infer Megan might be interested in sci-fi movies.

$$\begin{array}{c|ccccc} & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ \hline \text{Joe} & 1 & 1 & 1 & 0 & 0 \\ \text{Jim} & 3 & 3 & 3 & 0 & 0 \\ \text{John} & 4 & 4 & 4 & 0 & 0 \\ \text{Jack} & 5 & 5 & 5 & 0 & 0 \\ \text{Jill} & 0 & 0 & 0 & 4 & 4 \\ \text{Jenny} & 0 & 0 & 0 & 5 & 5 \\ \text{Jane} & 0 & 0 & 0 & 2 & 2 \end{array} = \begin{bmatrix} .14 & 0 \\ .42 & 0 \\ .56 & 0 \\ .70 & 0 \\ 0 & .60 \\ 0 & .75 \\ 0 & .30 \end{bmatrix} \begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix} \begin{bmatrix} .58 & .58 & .58 & 0 & 0 \\ 0 & 0 & 0 & .71 & .71 \end{bmatrix}$$

SVD utility: concept-space

But we can map Megan *back* into **movie-space**, too! Here, we'd be multiplying mV by V^T, which is the reverse of V (V^T maps movies $\leftarrow$ concepts). We find that $(mV)V^T =$ $\begin{bmatrix} 2.32 & 0 \end{bmatrix} \begin{bmatrix} .58 & .58 & .58 & 0 & 0 \\ 0 & 0 & 0 & .71 & .71 \end{bmatrix} = \begin{bmatrix} \underline{1.35 \quad 1.35 \quad 1.35} & 0 & 0 \end{bmatrix}$, so we can infer Megan might be interested Alien or Star Wars as much as she liked the Matrix.



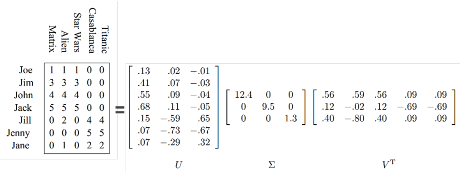
$$\begin{array}{c|ccccc|} & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ \hline \text{Joe} & 1 & 1 & 1 & 0 & 0 \\ \text{Jim} & 3 & 3 & 3 & 0 & 0 \\ \text{John} & 4 & 4 & 4 & 0 & 0 \\ \text{Jack} & 5 & 5 & 5 & 0 & 0 \\ \text{Jill} & 0 & 0 & 0 & 4 & 4 \\ \text{Jenny} & 0 & 0 & 0 & 5 & 5 \\ \text{Jane} & 0 & 0 & 0 & 2 & 2 \\ \hline \end{array} = \underbrace{\begin{bmatrix} .14 & 0 \\ .42 & 0 \\ .56 & 0 \\ .70 & 0 \\ 0 & .60 \\ 0 & .75 \\ 0 & .30 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} .58 & .58 & .58 & 0 & 0 \\ 0 & 0 & 0 & .71 & .71 \end{bmatrix}}_{V^{\mathrm{T}}}$$

SVD utility: concept-space

We can also map all users into concept-space and make recommendations by comparing their similarities there. (e.g., using cosine similarity)We can also map all users into concept-space and make recommendations by comparing their similarities there. (e.g., using cosine similarity)
Example: from the slightly more interesting example earlier:

Example: from the slightly more interesting example earlier:

$$MV = \begin{bmatrix} 1.71 & 0.22 & 0 \\ 5.13 & 0.66 & 0 \\ 6.84 & 0.88 & 0 \\ 8.55 & 1.1 & 0 \\ 1.9 & -5.56 & -0.88 \\ 0.9 & -6.9 & 0.9 \\ 0.95 & -2.78 & -0.44 \end{bmatrix}$$



$$\begin{array}{r|ccccc|} & \text{Matrix} & \text{Alien} & \text{Star Wars} & \text{Casablanca} & \text{Titanic} \\ \hline \text{Joe} & 1 & 1 & 1 & 0 & 0 \\ \text{Jim} & 3 & 3 & 3 & 0 & 0 \\ \text{John} & 4 & 4 & 4 & 0 & 0 \\ \text{Jack} & 5 & 5 & 5 & 0 & 0 \\ \text{Jill} & 0 & 2 & 0 & 4 & 4 \\ \text{Jenny} & 0 & 0 & 0 & 5 & 5 \\ \text{Jane} & 0 & 1 & 0 & 2 & 2 \\ \hline \end{array} = \underbrace{\begin{bmatrix} .13 & .02 & -.01 \\ .41 & .07 & -.03 \\ .55 & .09 & -.04 \\ .68 & .11 & -.05 \\ .15 & -.59 & .65 \\ .07 & -.73 & -.67 \\ .07 & -.29 & .32 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} .56 & .59 & .56 & .09 & .09 \\ .12 & -.02 & .12 & -.69 & -.69 \\ .40 & -.80 & .40 & .09 & .09 \end{bmatrix}}_{V^{\mathrm{T}}}$$

Computing SVD

So how do we actually *do* this?

SVD: $U\Sigma V^T$

Where Σ is a diagonal matrix full of A's singular values σ_i

Eigenvalue decomposition: for a symmetric matrix A, $A = X\Lambda X^T$

Where Λ is a diagonal matrix full of A's eigenvalues λ_i

Both: U, V, X are orthonormal

Question: If we write A using its SVD, what is AA^T? And what is A^TA?

Computing SVD

So how do we actually *do* this?

SVD: $U\Sigma V^T$

Where Σ is a diagonal matrix full of A's singular values σ_i

Eigenvalue decomposition: for a symmetric matrix A, $A = X\Lambda X^T$

Where Λ is a diagonal matrix full of A's eigenvalues λ_i

Both: U, V, X are orthonormal

Question: If we write A using its SVD, what is AA^T? And what is A^TA?

- $AA^T = U\Sigma V^T(U\Sigma V^T)^T = U\Sigma V^T V\Sigma^T U^T = U\Sigma\Sigma^T U^T$
- $A^TA = V\Sigma^2 V^T$, similarly
- **Result:** the singular values *squared* Σ^2 are the eigenvalues of both AA^T and A^TA.

Computing SVD

Result: the singular values *squared* Σ^2 are the eigenvalues of both AA^T and A^TA. And the matrices U and V?

We have: $AA^T = U\Sigma^2U^T$ and $A^TA = V\Sigma^2V^T$

- First, Multiply AA^T on the right by U:

$$AA^TU = U\Sigma^2U^TU = U\Sigma^2$$

... or U's columns are *eigenvectors* of AA^T

- Similarly, multiply A^TA on the right by V:

$$A^TAV = V\Sigma^2V^TV = V\Sigma^2$$

... or V's columns are eigenvectors of A^TA

SVD Algorithm

To compute the SVD of a matrix A:

1. Construct AA^T and A^TA
2. Use the generalized power iteration algorithm to compute the eigenvalues and eigenvectors of each (or something faster, maybe trust in your software? Or take an advanced numerics class!)
   (Eigenvalues should be the same! But numerical imprecision.... maybe not :)
3. Σ= diag(square roots of the eigenvalues of AA^T or A^TA, in descending order)
4. U = matrix whose columns are the eigenvectors of AA^T (in descending order by their associated eigenvalue)
5. V= matrix whose columns are the eigenvectors of ATA (in descending order by their associated eigenvalue)

1. Construct AA^T and A^TA
2. Use the generalized power iteration algorithm to compute the eigenvalues and eigenvectors of each (or something faster, maybe trust in your software? Or take an advanced numerics class!)
   (Eigenvalues should be the same! But numerical imprecision.... maybe not.)
3. $\Sigma=$ diag(square roots of the eigenvalues of AA^T or A^TA, in descending order)
4. $U =$ matrix whose columns are the eigenvectors of AA^T (in descending order by their associated eigenvalue)
5. $V=$ matrix whose columns are the eigenvectors of A^TA (in descending order by their associated eigenvalue)

1. Construct AA^T and A^TA
2. Use the generalized power iteration algorithm to compute the eigenvalues and eigenvectors of each (or something faster, maybe trust in your software? Or take an advanced numerics class!)
   (Eigenvalues should be the same! But numerical imprecision.... maybe not.)
3. $\Sigma =$ diag(square roots of the eigenvalues of AA^T or A^TA, in descending order)
4. $U =$ matrix whose columns are the eigenvectors of AA^T (in descending order by their associated eigenvalue)
5. $V =$ matrix whose columns are the eigenvectors of A^TA (in descending order by their associated eigenvalue)
   To reduce dimension:
6. Keep a running total of the sum of squared singular values (as an array, will naturally be in descending order).
7. Once you have all of them, divide the whole array by the total, see where 80-90%; zero out all the singular values after this point!

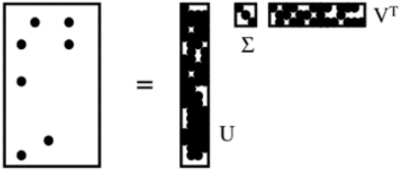
SVD: Pros and Cons

Pros:

1. Optimal low-rank approximation
2. Always exists, for any matrix

Cons:

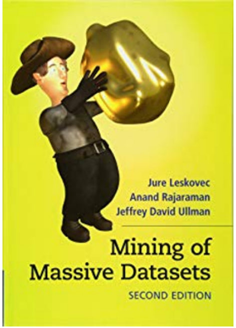
1. Interpretability problem
2. A singular vector specifies a linear combination of all input columns/rows – what does this mean?
3. Lack of sparsity
4. Our original matrix may have been sparse (users/movies/ratings, e.g., certainly is!), but U and V have lots of nonzero entries



Acknowledgments

Next time: more matrix decompositions

Some material is adapted/adopted from Mining of Massive Data Sets, by Jure Leskovec, Anand Rajaraman, Jeff Ullman (Stanford University) `http://www.mmds.org`



Special thanks to Tony Wong for sharing his original adaptation and adoption of slide material.