Math Lecture 5

Linear algebra tools for data science

- Now we’ve seen some of the basic building blocks of linear algebra—let’s see how they come together for some of the common matrix tools for data science

Outline for today

- Principal component analysis
- Matrix decompositions
  - Eigendecomposition
  - Singular value decomposition (SVD)
  - Other matrix decompositions (LU, QR, etc.)

Principal Component Analysis

Main idea: use eigenvalues and eigenvectors to understand the main directions of variation in the data

Uses

- Understanding the structure of (high dimensional) data
- Dimension reduction (to help with computation, data visualization, etc)
- Data normalization/alignment

How does it work?

Covariance matrix

- Matrix of the variance and covariances of each variable of the data
- Describes the variation of each variable with respect to each other
- Sends each standard basis (cardinal direction) vector to the variance and covariances of that direction in the data (i.e. of that variable)

$$\begin{bmatrix} \mathrm{var}(x_1) & \mathrm{cov}(x_1,x_2) & \cdots & \mathrm{cov}(x_1,x_n) \\ \mathrm{cov}(x_2,x_1) & \mathrm{var}(x_2) & & \vdots \\ \vdots & & \ddots & \\ \mathrm{cov}(x_n,x_1) & \cdots & & \mathrm{var}(x_n) \end{bmatrix}$$

🟩 variances

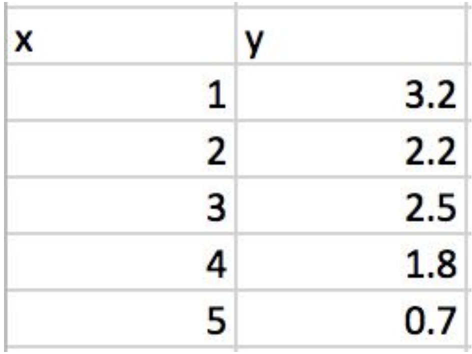
🟦 covariances

$$\mathrm{Var}(X) = \mathrm{E}\left[(X-\mu)^2\right].$$

$$\mathrm{cov}(X,Y) = \mathrm{E}\left[(X-\mathrm{E}[X])(Y-\mathrm{E}[Y])\right]$$

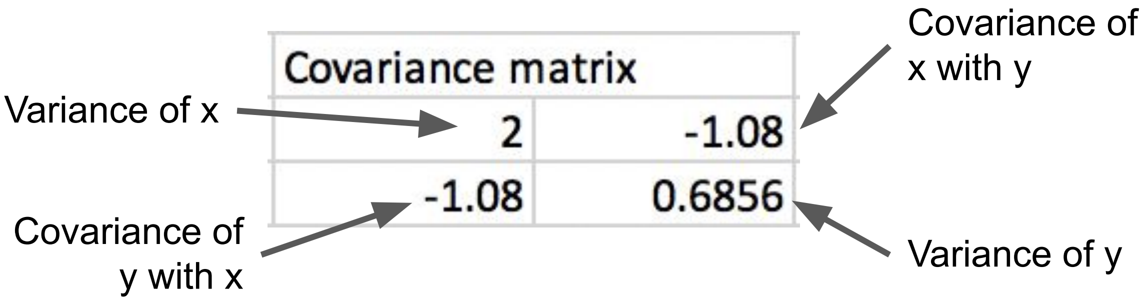
Example

Data



x	y
1	3.2
2	2.2
3	2.5
4	1.8
5	0.7

Mean x: 3
Mean y: 2.08



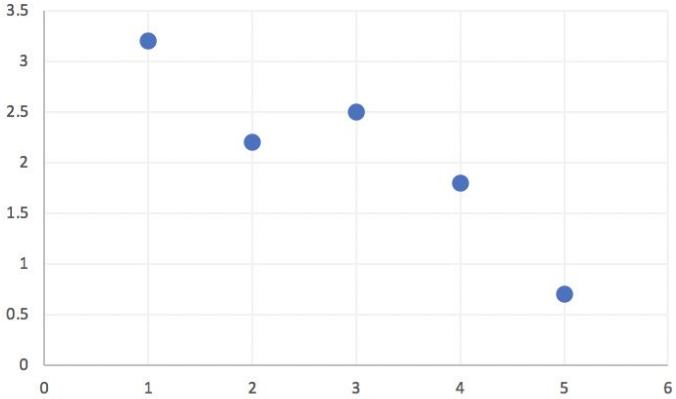
Covariance matrix	
2	-1.08
-1.08	0.6856

Variance of x

Covariance of x with y

Covariance of y with x

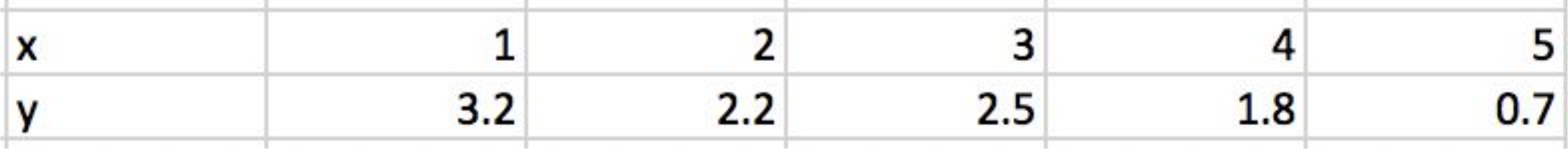
Variance of y



- How does the covariance matrix relate to the trend in the data?
- Does one variable have more variation than the other?

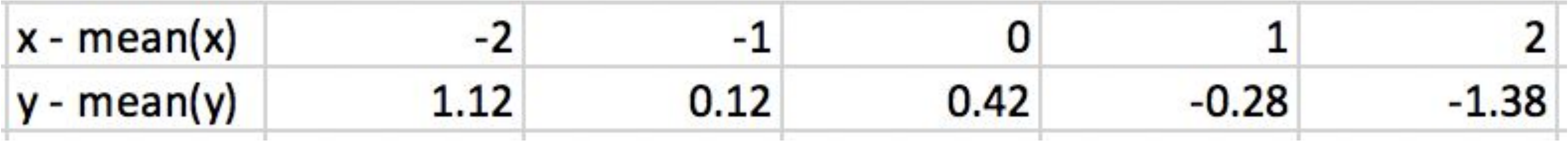
Fun problem to think about—work out why this is true:

Say we have a data set, and we write it so each data point forms a column vector:



x	1	2	3	4	5
y	3.2	2.2	2.5	1.8	0.7

Then, if we recenter the data around it’s mean:



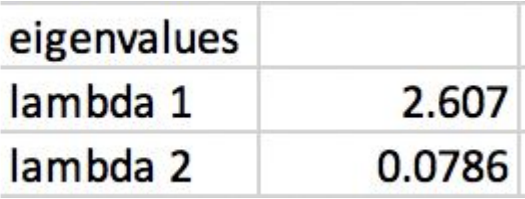
x - mean(x)	-2	-1	0	1	2
y - mean(y)	1.12	0.12	0.42	-0.28	-1.38

Call this matrix A. Suppose we have n data points (n columns).

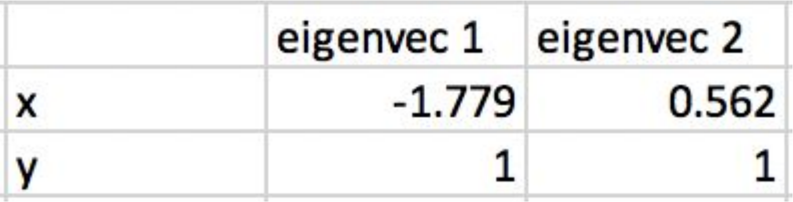
Then Cov(data) = AA^T/n (and if we did the matrix with data points as rows and variables as columns, then Cov(data) = A^TA/n)

Eigenvalues and eigenvectors of the covariance matrix

- Gives a sense of the scale and direction of variation in the data
- Eigenvectors tell us directions of variation, and the eigenvalue for each eigenvector tells us how much of the variation is in that direction

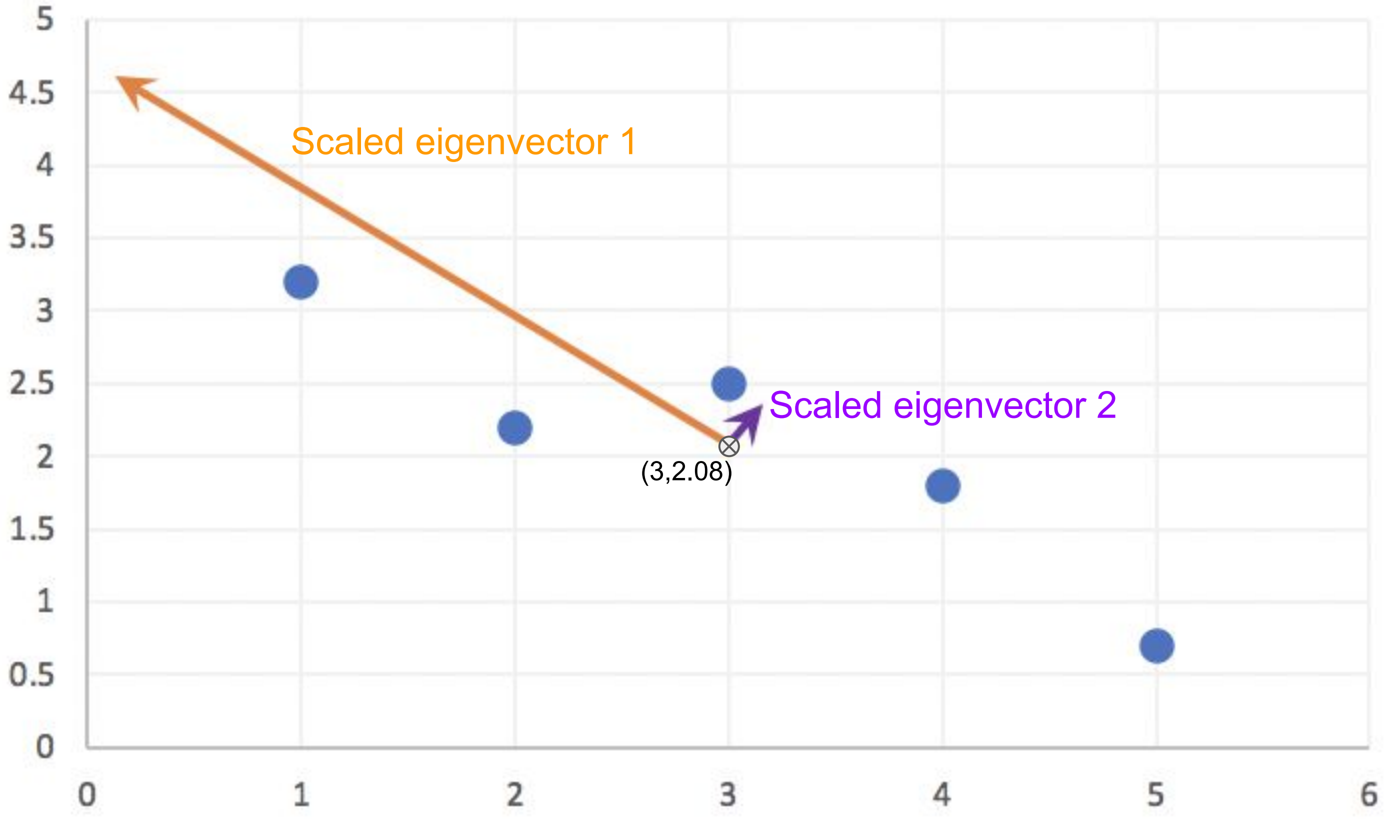


eigenvalues	
lambda 1	2.607
lambda 2	0.0786



	eigenvec 1	eigenvec 2
x	-1.779	0.562
y	1	1

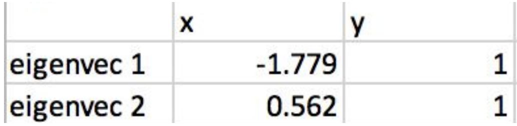
- Position the the eigenvectors at the mean of the data (mean(x) = 3, mean(y) = 2.08), and scale them to the size of the eigenvalues
- Actually want to scale to the square root of the eigenvalues—the eigenvalues tell us about the variance in the data in that direction, so if we take the square root we get the standard deviation



Scaled eigenvector 1
Scaled eigenvector 2
(3,2.08)
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
1
2
3
4
5
6

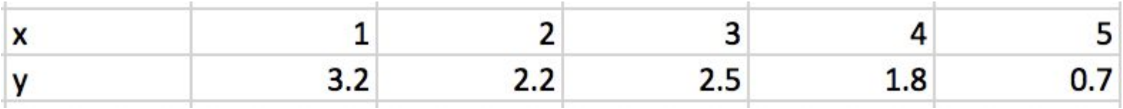
Rotate the data to match the directions of the eigenvectors

If we multiply our data by the matrix of our eigenvectors, it transforms the data so the cardinal directions are the eigenvectors (change of basis):

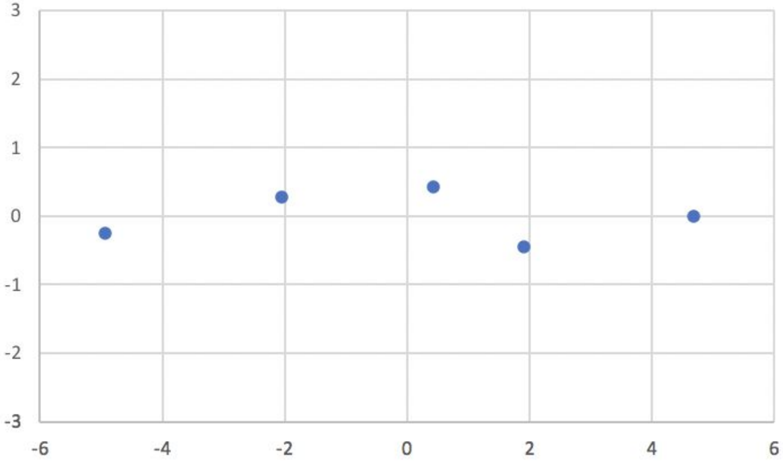


	x	y
eigenvec 1	-1.779	1
eigenvec 2	0.562	1

X



x	1	2	3	4	5
y	3.2	2.2	2.5	1.8	0.7



Now the x direction represents distance on eigenvector 1 and y direction represents distance on eigenvector 2

How do the eigenvectors help us understand the data?

- We may not need the full high dimensional data set—e.g. this data set can probably be mostly explained with just one dimension!

- Rotating onto the eigenvector directions lets us normalize/align the data

- We can use this to uncover patterns in the data—the eigenvectors may correspond to features in the data that are meaningful

Another example

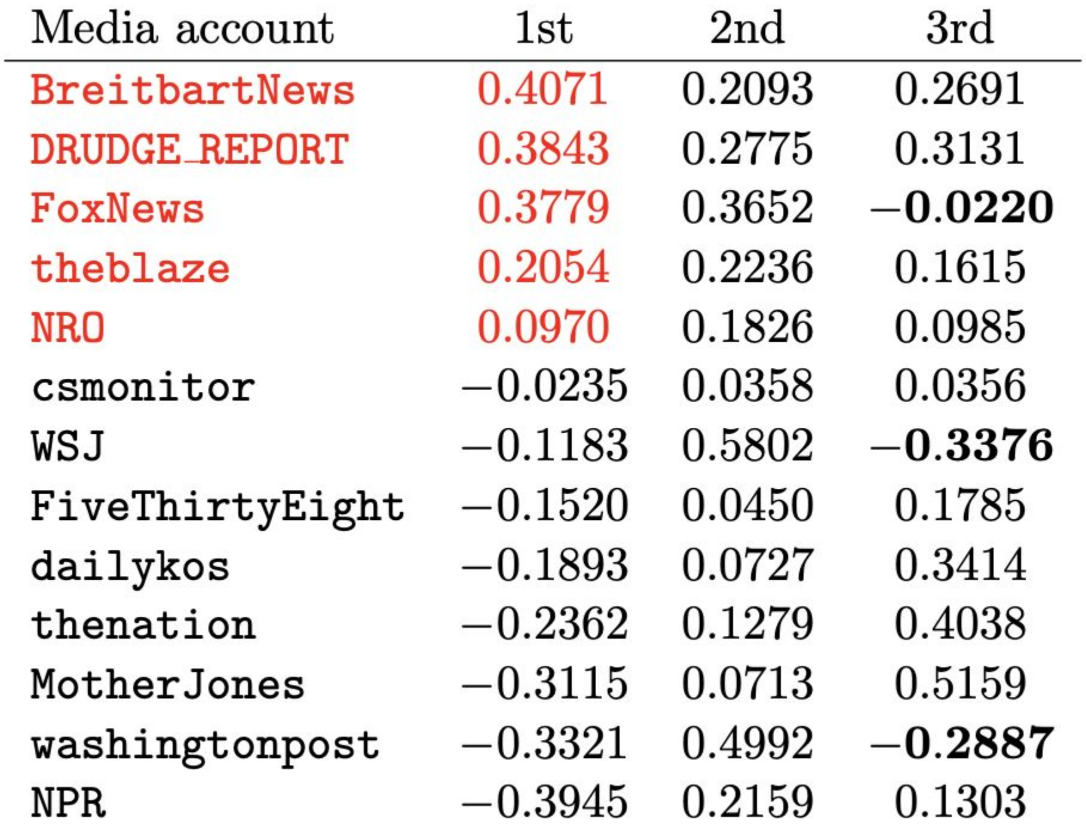
- Code notebook together!

https://colab.research.google.com/drive/1m-OymabBzYfuR3AGUCqi2n9F86bjQpSg#scrollTo=8VOeuWzIDfLM

Twitter example

Twitter data on media followership

- Each row is a twitter account
- Columns are 0's and 1's for whether they follow each account
- 99,412 accounts tracked



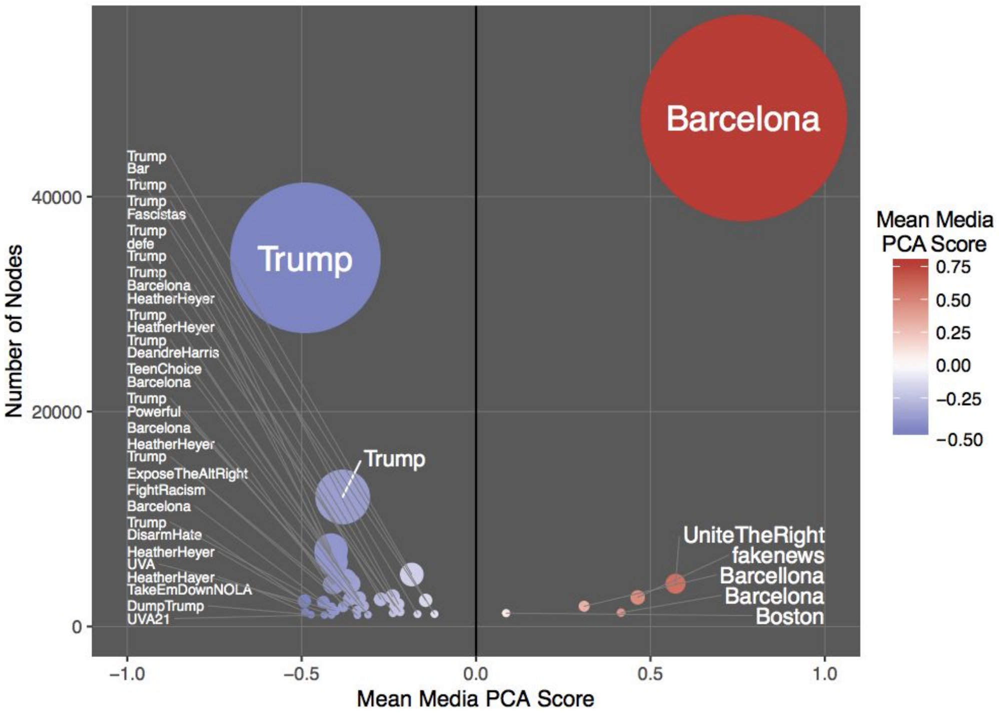
Media account	1st	2nd	3rd
BreitbartNews	0.4071	0.2093	0.2691
DRUDGE_REPORT	0.3843	0.2775	0.3131
FoxNews	0.3779	0.3652	**−0.0220**
theblaze	0.2054	0.2236	0.1615
NRO	0.0970	0.1826	0.0985
csmonitor	−0.0235	0.0358	0.0356
WSJ	−0.1183	0.5802	**−0.3376**
FiveThirtyEight	−0.1520	0.0450	0.1785
dailykos	−0.1893	0.0727	0.3414
thenation	−0.2362	0.1279	0.4038
MotherJones	−0.3115	0.0713	0.5159
washingtonpost	−0.3321	0.4992	**−0.2887**
NPR	−0.3945	0.2159	0.1303

https://arxiv.org/pdf/1905.07755.pdf

Use the 1st principal component to assign a media score

Explored different communities on twitter, the media they follow and what they tweeted about the 2017 ‘Unite the Right’ rally in Charlottesville

Uses linear algebra to partition the network into communities as well—more on this later in the semester!



Most common hashtags used by different communities following the Unite the Right rally

https://arxiv.org/pdf/1905.07755.pdf

Matrix Decomposition

What is a matrix decomposition?

- A way to break or factorize matrices into different pieces or sub-matrices
- Often easier to do computations with the different pieces, so matrix decomposition gets used in a wide range of algorithms
- For large, high dimensional data, you can use matrix decomposition to find smaller simpler ways to represent the data (e.g. data compression)

- Lots of different decompositions out there, we will explore a couple here!

Eigendecomposition

- Gives us a way to represent a matrix in terms of its eigenvalues and eigenvectors

Switch to ipad!