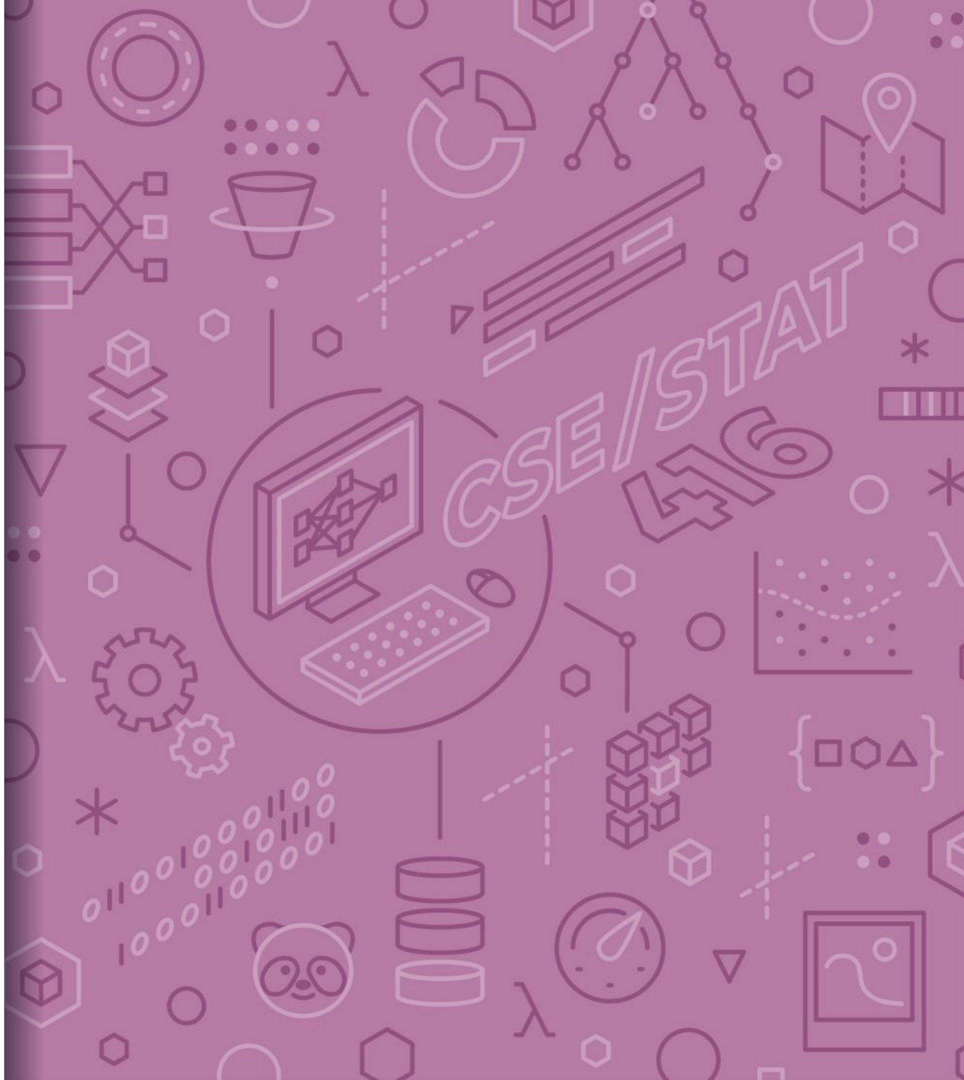
CSE/STAT 416

Dimensionality Reduction & Recommender Systems Intro

Amal Nanavati
University of Washington
Aug 8, 2022

Adapted from Hunter Schafer’s slides



Administrivia

- **This Week**: Dimensionality Reduction, Recommender Systems
- **Next Week**: Course Wrap-Up & Guest Panel
- Deadlines:
  - HW6 due TOMORROW, Tues 8/9 11:59PM
    - Submit Concept on Gradescope
    - Submit Programming on EdSTEM
  - HW7 (final HW) released Wed 8/10
    - Due Tues 8/16 11:59PM, **NO LATE DAYS**
  - LR 8 due Fri 8/12 11:59PM
  - **Extra Credit** Guest Panel Mon 8/15 during lecture.
  - Take-Home Final Exam: Wed 8/17 – Thurs 8/18

9AM 11:59PM

Addressing LR Questions

Fairness Definitions

1. “Fairness through Unawareness”
    1. To avoid unfair decisions, prevent the model from every looking at protected attribute (e.g., race, gender).
    2. **Doesn’t work in practice**
2. Statistical Parity
    1. Idea: Equal performance across groups.

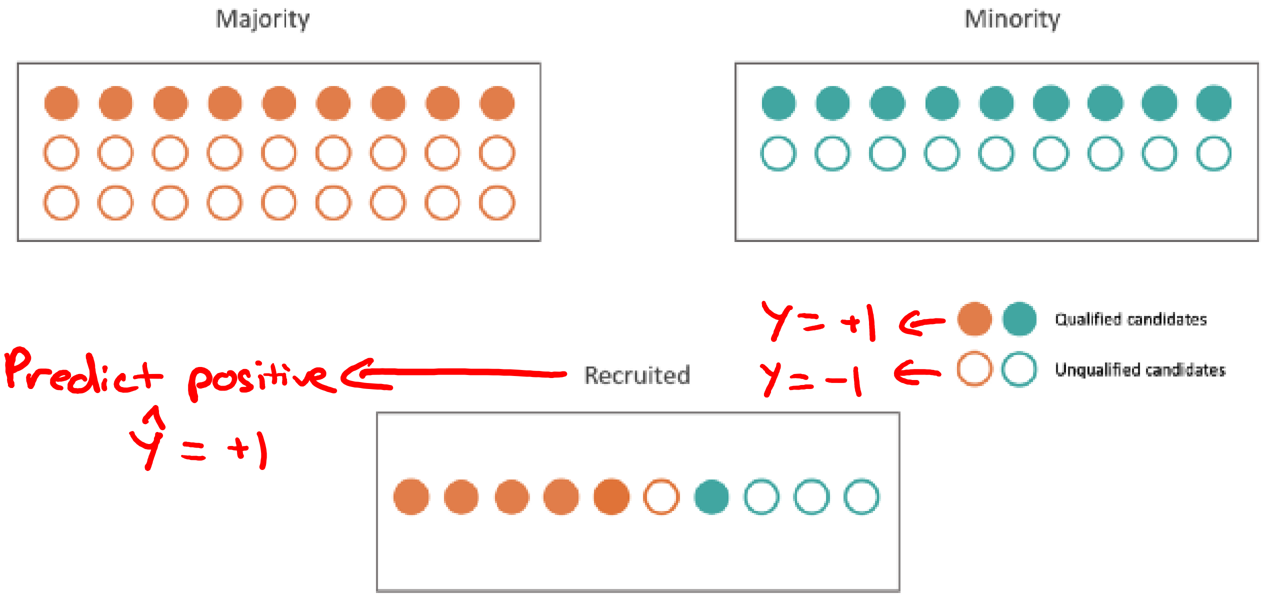
prediction

$$\Pr\left(\hat{Y} = +|\ A = ■\right) = \Pr\left(\hat{Y} = +|\ A = ○\right)$$

    2. Also phrased as matching demographic statistics (e.g., if 33% of population are Circles, 33% of those admitted should be Circles).
3. Equal Opportunity
    1. Idea: True positive rate should be equal across groups

$$\Pr\left(\underset{\substack{\downarrow\\ \text{predictions}}}{\hat{Y}} = +|\ A = ■, \underset{\substack{\downarrow\\ \text{Actual labels}}}{Y} = +\right) = \Pr\left(\hat{Y} = +|\ A = ○, Y = +\right)$$

Statistical Parity Example



Majority

Minority

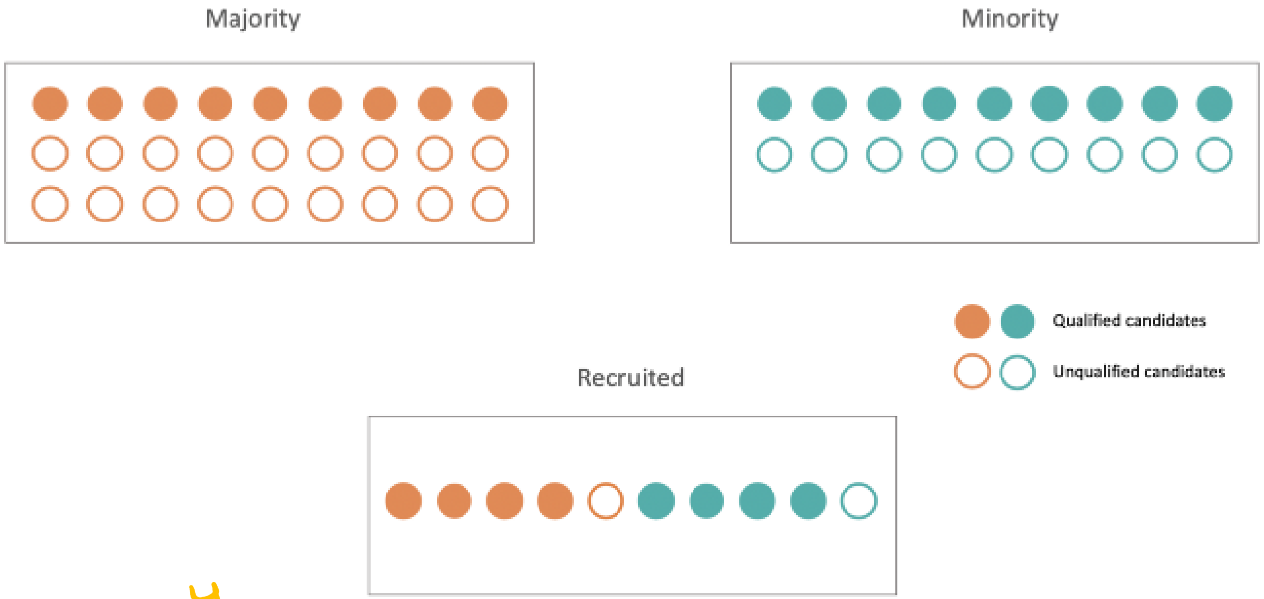
$Y = +1$ ← ● ● Qualified candidates

$Y = -1$ ← ○ ○ Unqualified candidates

Predict positive ← Recruited

$\hat{Y} = +1$

Equality of Opportunity Example

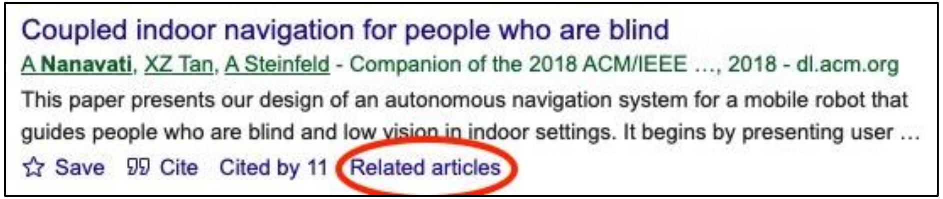


$TPR = \frac{4}{9}$

$TPR = \frac{4}{9}$

Unsupervised Learning

- A type of machine learning that detects underlying patterns in **unlabeled** data.
- Examples of unsupervised learning tasks:
  - Cluster similar articles together.

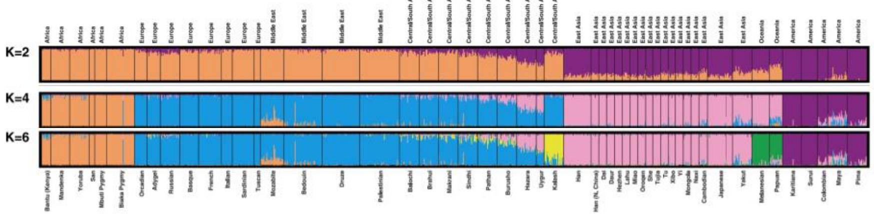


Coupled indoor navigation for people who are blind

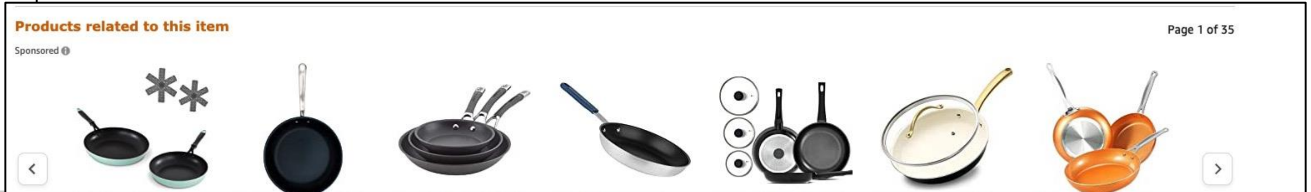
A Nanavati, XZ Tan, A Steinfeld - Companion of the 2018 ACM/IEEE …, 2018 - dl.acm.org

This paper presents our design of an autonomous navigation system for a mobile robot that guides people who are blind and low vision in indoor settings. It begins by presenting user …

  - Cluster gene sequences.



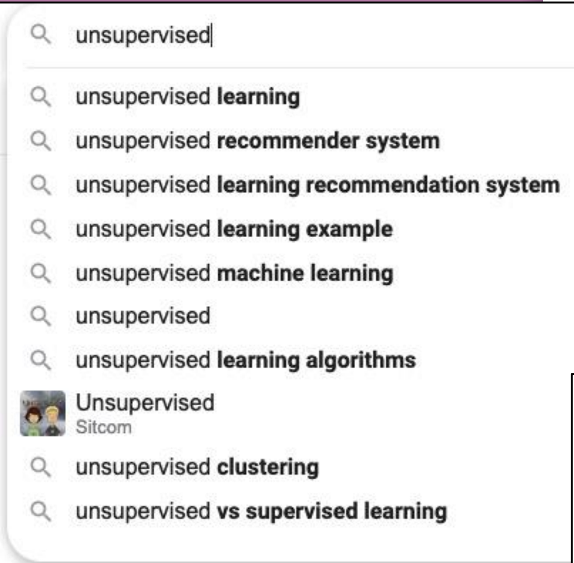
  - Recommend items, searches, movies, etc.



Products related to this item

Sponsored ⓘ

Page 1 of 35



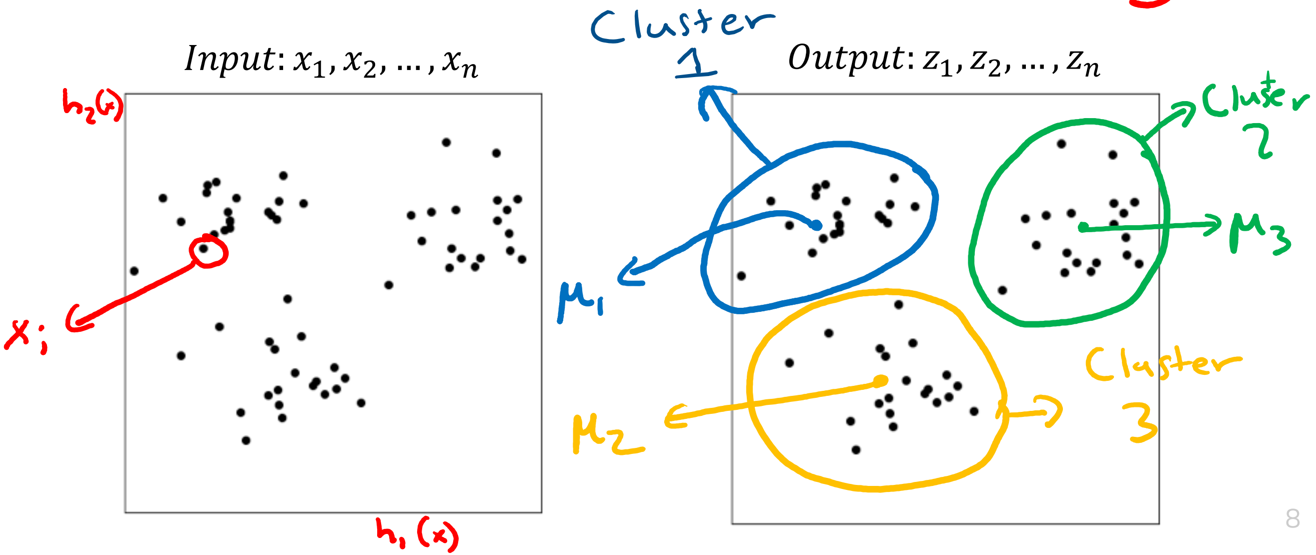
unsupervised

- unsupervised **learning**
- unsupervised **recommender system**
- unsupervised **learning recommendation system**
- unsupervised **learning example**
- unsupervised **machine learning**
- unsupervised
- unsupervised **learning algorithms**
- Unsupervised
  Sitcom
- unsupervised **clustering**
- unsupervised **vs supervised learning**

Unlabeled Data

- In many real world contexts, there aren't clearly defined labels so we won't be able to do classification
- We will need to come up with methods that uncover structure from the (unlabeled) input data X.
- **Clustering** is an automatic process of trying to find related groups within the given dataset.

$z_i \in [1, 2, 3]$



Poll Everywhere

Think

2 min

- Which word(s) have the largest IDF? Which word(s) have the smallest IDF?

Red = low

Green = high

Review
“Sushi was great, the food was awesome, but the service was terrible”
“Terrible food; the sushi was rancid.”

Note that if we divide the Bag of Words embedding by the num words in the document, we get the TF!

Sushi	was	great	the	food	awesome	but	service	terrible	rancid
1	3	1	2	1	1	1	1	1	0
1	1	0	1	1	0	0	0	1	1

Coordinate Descent

k-means is trying to minimize the heterogeneity objective

$$\underset{\underline{z},\underline{\mu}}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{i=1}^{n} \mathbf{1}\{z_i = j\} \left|\left|\mu_j - x_i\right|\right|_2^2$$

Step 0: Initialize cluster centers

Repeat until convergence: fix μ, minimize z

Step 1: Assign each example to its closest cluster centroid

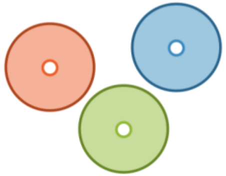
Step 2: Update the centroids to be the mean of all the points assigned to that cluster fix z, minimize μ

Coordinate Descent alternates how it updates parameters to find minima. On each of iteration of Step 1 and Step 2, heterogeneity decreases or stays the same.

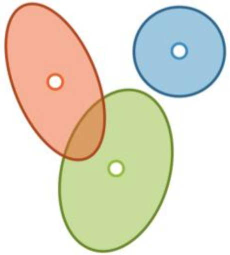
=> Will converge in finite time

Finding Shapes

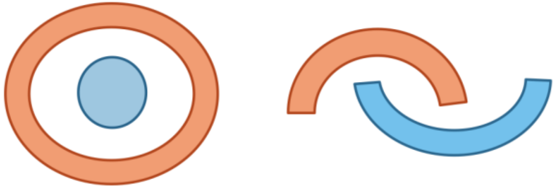
k-means



Mixture Models



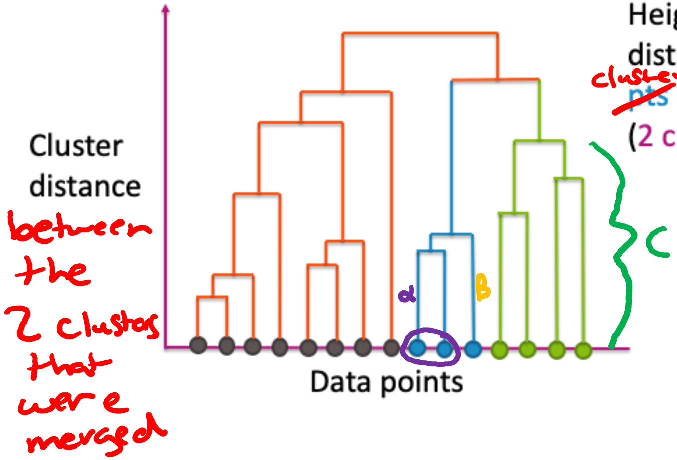
Hierarchical Clustering



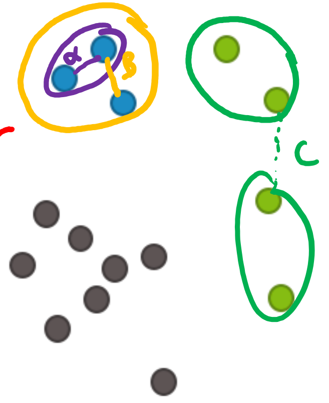
Dendrogram

x-axis shows the datapoints (arranged in a very particular order)

y-axis shows distance between pairs of clusters



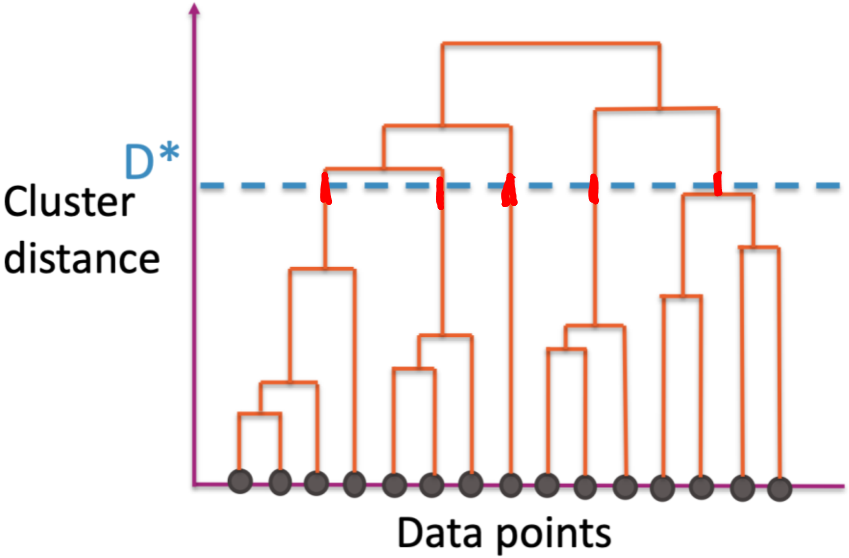
Height here indicates min distance between blue ~~pts~~ cluster and green ~~pts~~ cluster (2 clusters)



Cut Dendrogram

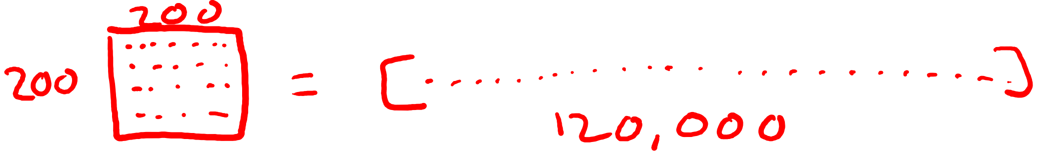
Choose a distance D^* to “cut” the dendrogram

- Use the largest clusters with distance < D^*
- Usually ignore the idea of the nested clusters after cutting



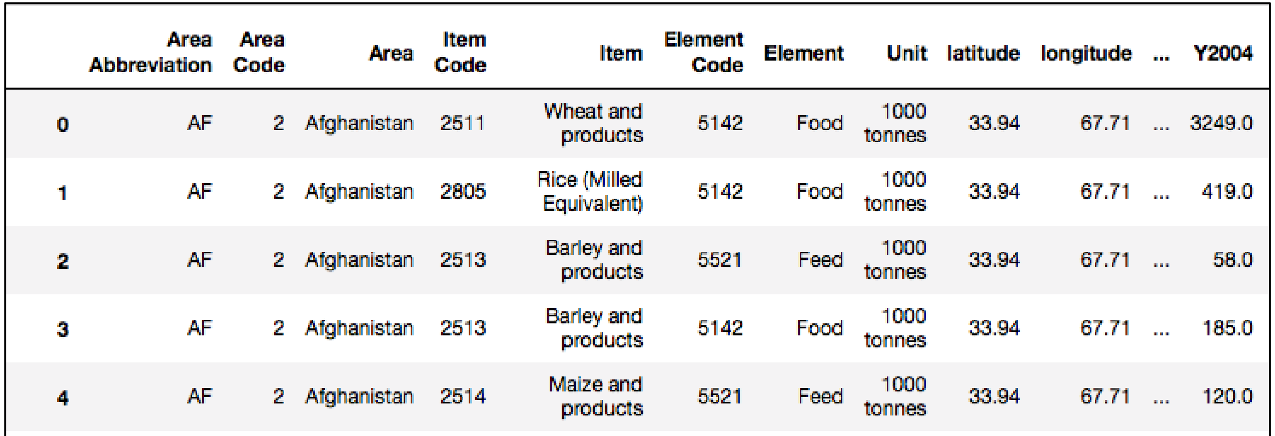
Dimensionality Reduction

Large Dimensionality



Input data might have thousands or millions of dimensions!

- **Images**: 200x200 image is 120,000 features!
- **Text**: # features = # n-grams 😮
- **Course Success**: dozen(s) of features HW 4
- **User Ratings**: 100s of ratings (one per rate-able item)

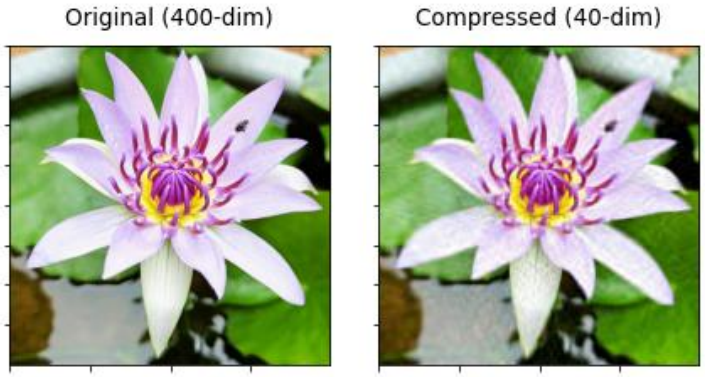


	Area Abbreviation	Area Code	Area	Item Code	Item	Element Code	Element	Unit	latitude	longitude	...	Y2004
0	AF	2	Afghanistan	2511	Wheat and products	5142	Food	1000 tonnes	33.94	67.71	...	3249.0
1	AF	2	Afghanistan	2805	Rice (Milled Equivalent)	5142	Food	1000 tonnes	33.94	67.71	...	419.0
2	AF	2	Afghanistan	2513	Barley and products	5521	Feed	1000 tonnes	33.94	67.71	...	58.0
3	AF	2	Afghanistan	2513	Barley and products	5142	Food	1000 tonnes	33.94	67.71	...	185.0
4	AF	2	Afghanistan	2514	Maize and products	5521	Feed	1000 tonnes	33.94	67.71	...	120.0

Issues with Too Many Dimensions

- **Visualization**: Hard to visualize more than 3D.
- **Overfitting**: Greater risk of overfitting with more features/dimensions

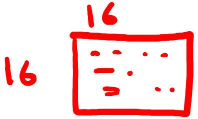
  supervised
  unsupervised
- **Scalability**: some ML approaches (e.g., k-nn, k-means) perform poorly in high-dimensional spaces (curse of dimensionality)
- **Redundancy**: high-dimensional data often occupies a lower-dimensional subspace.
  - Most pixels in MNIST (digit recognition) are white – are they necessary? only need 5 dimensions
    - Image Compression


Original (400-dim)

Compressed (40-dim)

Dimensionality Reduction is the the task of representing the data with a fewer number of dimensions, while keeping meaningful relations between data

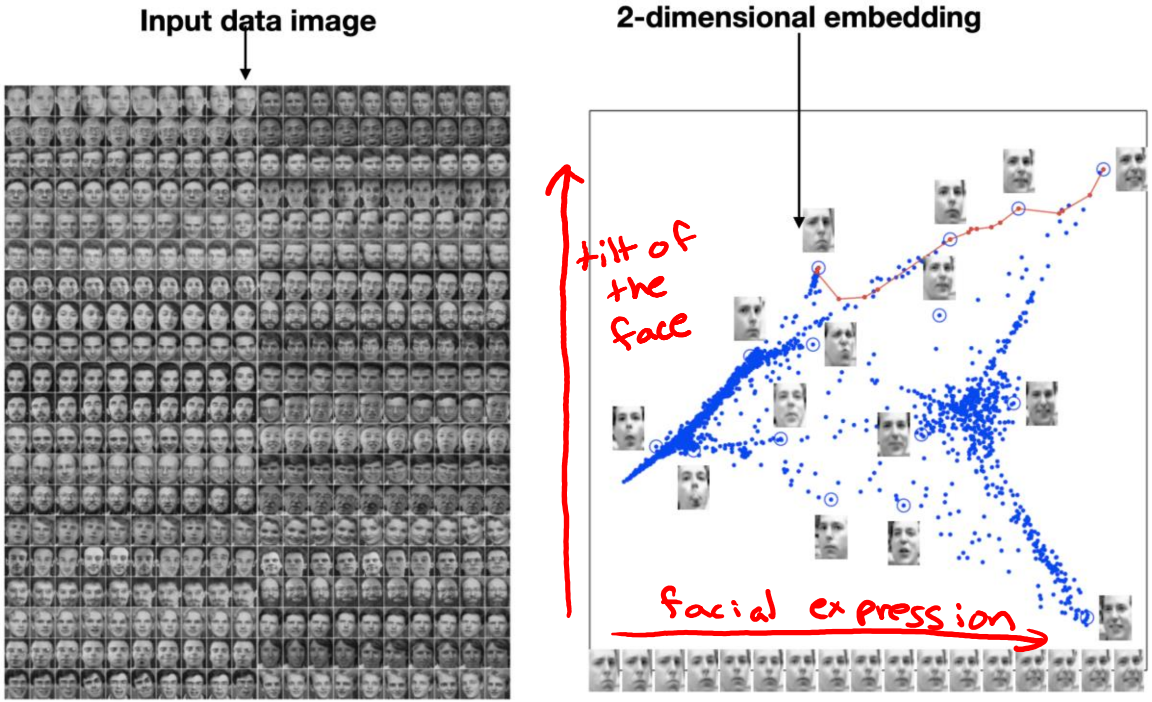
Example: Embedding Pictures



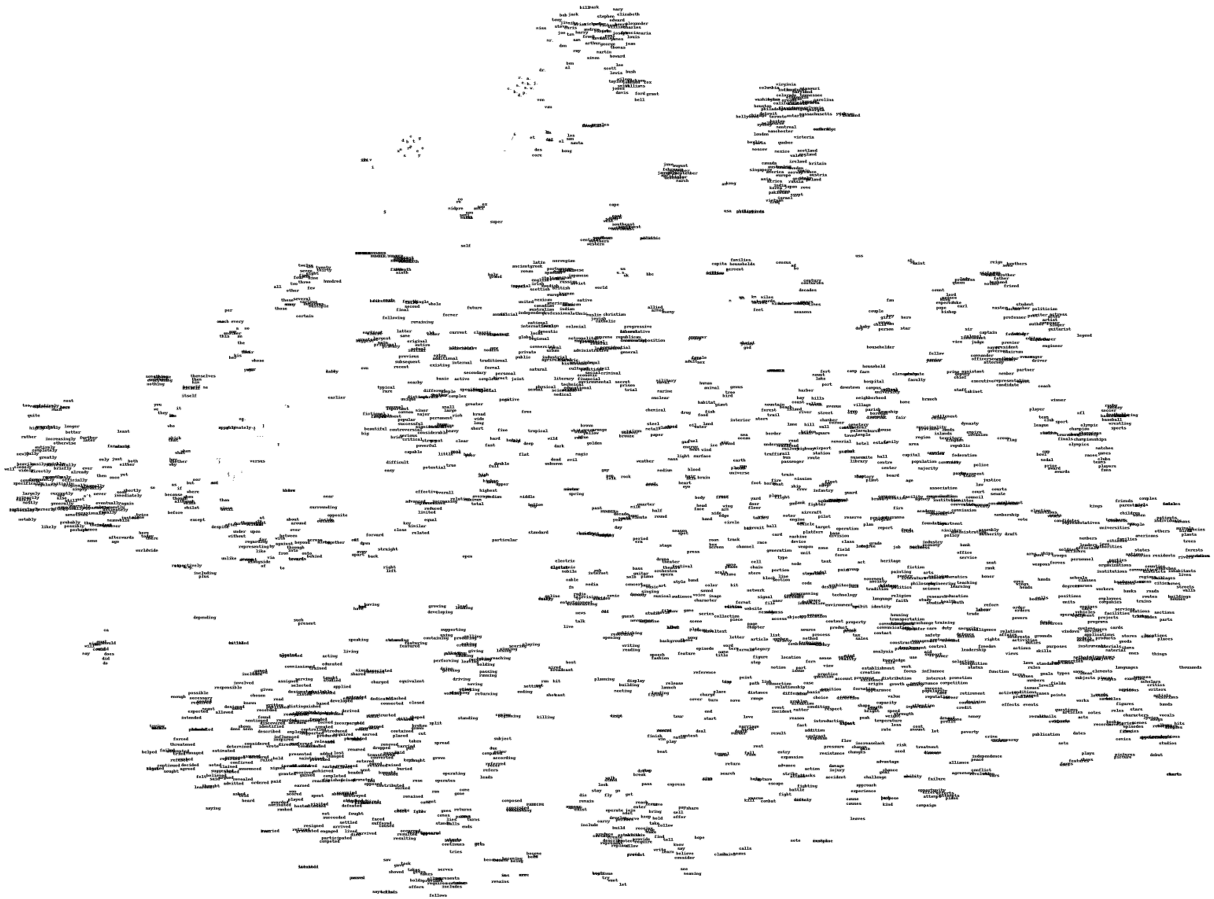
= 256 dimensions

Example: Embed high dimensional data in low dimensions to visualize the data

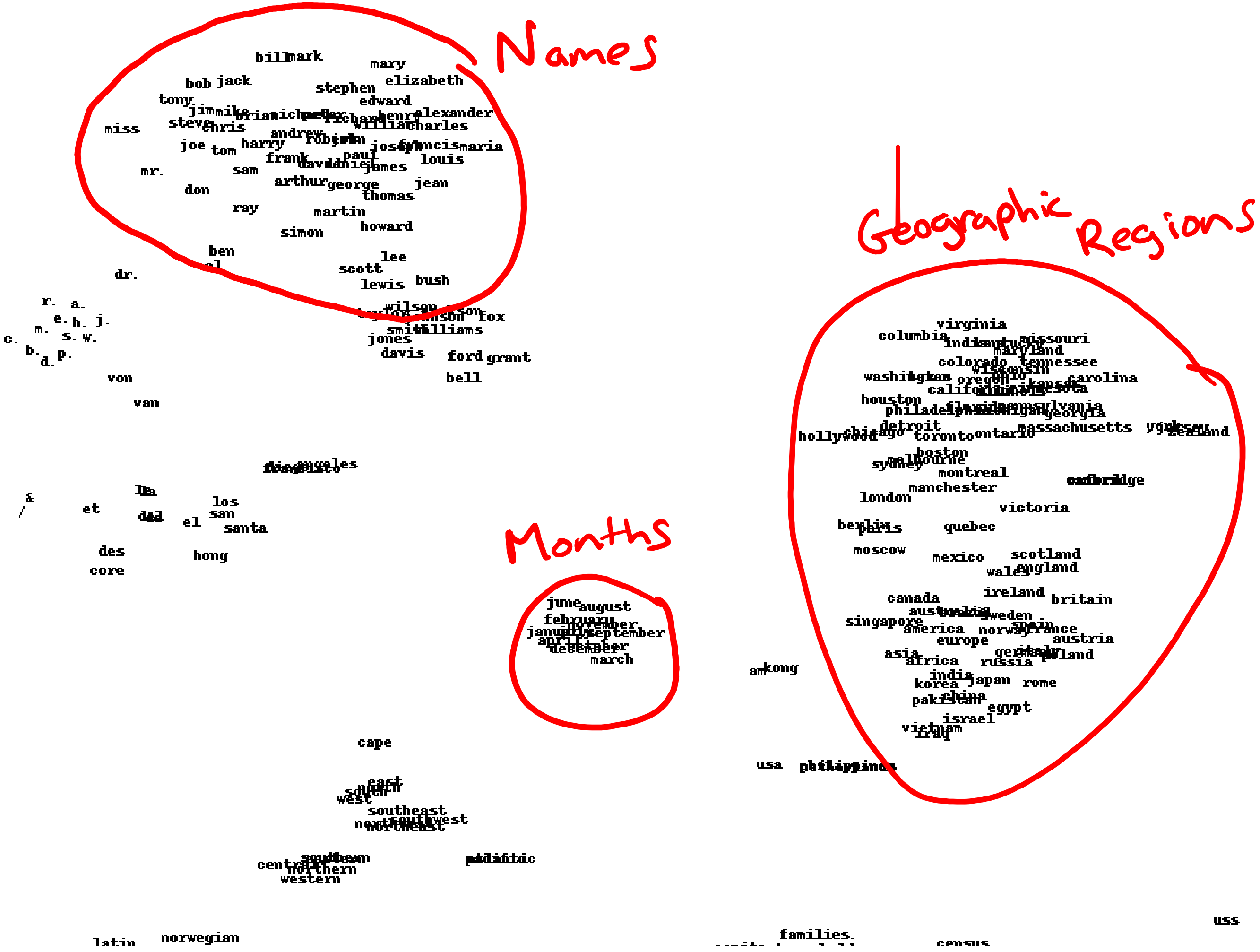
- Goal: Similar images should be near each other.



Example: Embedding Words



Example: Embedding Words



Principal Component Analysis (PCA)

One very popular dimensionality reduction algorithm is called **Principal Component Analysis (PCA)**.

?

Idea: Use a linear projection from d-dimensional data to k-dimensional data $k << d$

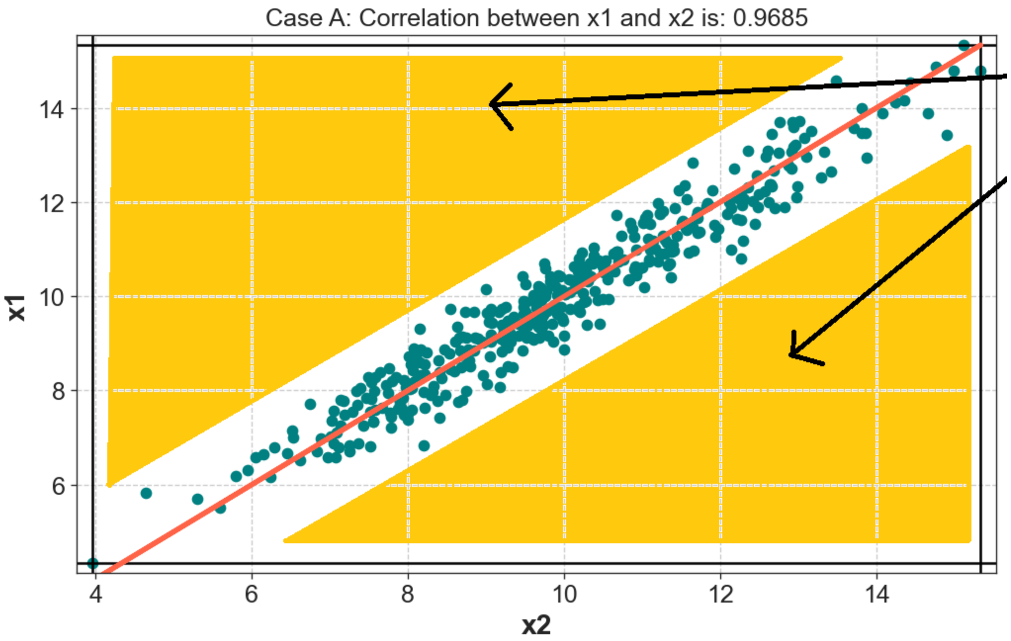
- E.g. 1000 dimension word vectors to 3 dimensions

?

Choose the projection that minimizes **reconstruction error**

- Idea: The information lost if you were to ”undo” the projection

Principal Component Analysis (PCA)

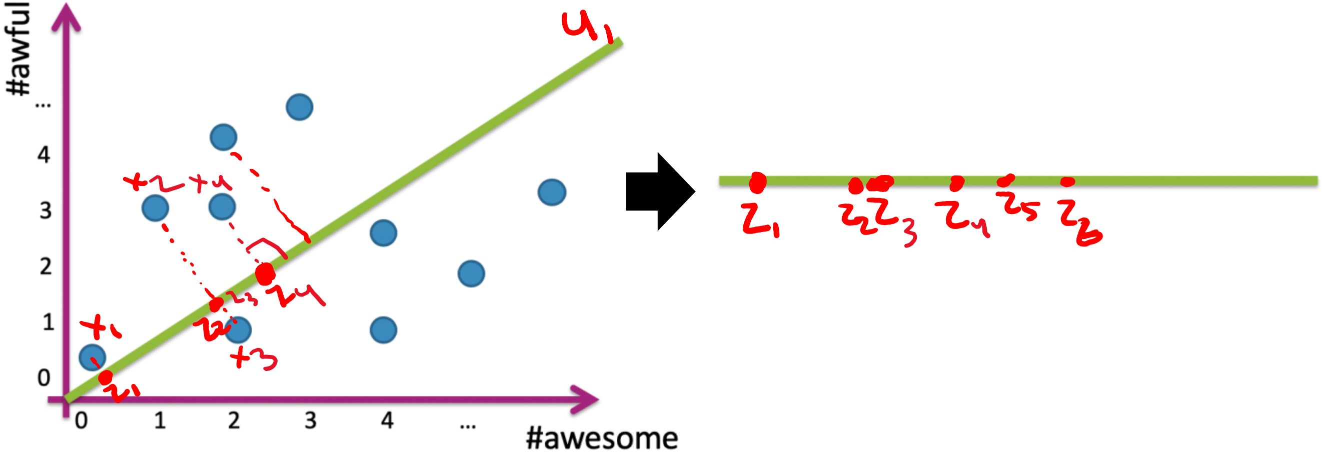


Regions with no data. Data exists close to a lower-dimensional subspace.

Linear Projection

Linear Projection of $\vec{x}_1$ onto $\vec{u}_1$ is the point on $\vec{u}_1$ that is closest to $\vec{x}_1$

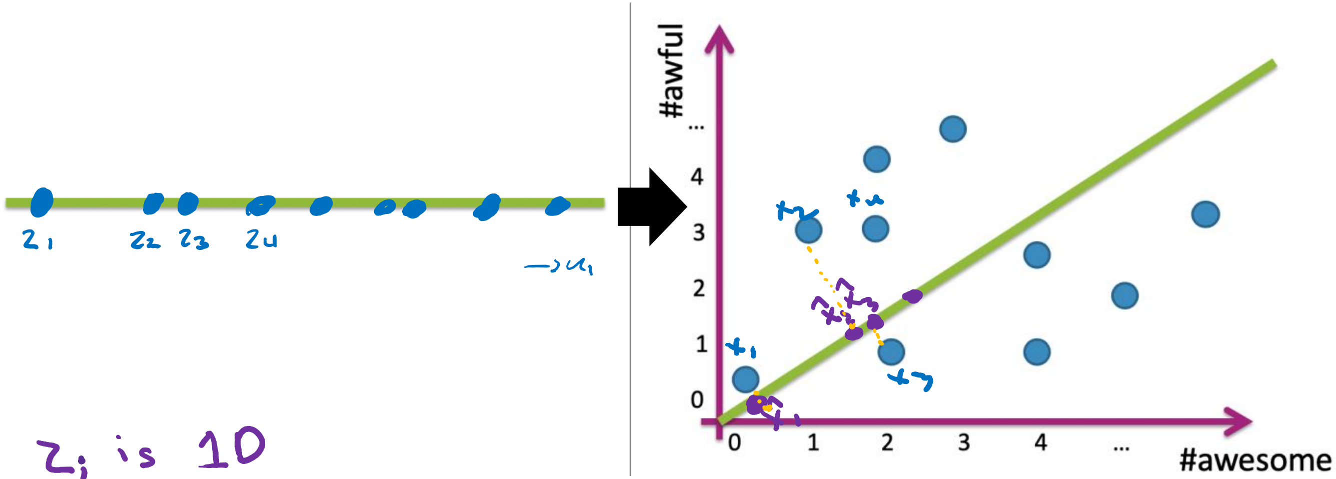
Project data into 1 dimension along a line



$$z_i = u_i^T x_i = \sum_{j=1}^{D} u_i[j] \cdot x_i[j]$$

Reconstruction

Reconstruct original data only knowing the projection



z_i is 1D

$\vec{u}_1$ is 2D vector

$\hat{x}_i = z_i u_i$

Reconstruction Error

$\|\hat{x}_i - x_i\|_2^2$

Poll Everywhere

Think

1 min

$x_i \in \mathbb{R}^5$

$z_i \in \mathbb{R}^2$

$z_{i,1} = \sum_{j=1}^{D} u_1[j] \cdot x_i[j]$

$z_{i,2} = \sum_{j=1}^{D} u_2[j] \cdot x_i[j]$

- Compute the 2D coordinates of the following point. Then compute its reconstruction error.
  - $x_i = [0, -7, 3, 2, 5]$
  - $u_1 = [-0.5, 0, 0.5, -0.5, 0.5]$
  - $u_2 = [0.5, 0, 0.5, -0.5, -0.5]$
  - $z_i = ??$ $(\underline{z_{i,1}}, \underline{z_{i,2}})$
  - $\hat{x}_i = ??$ $(\_\_, \_\_, \_\_, \_\_, \_\_)$
  - $||\hat{x}_i - x_i||_2^2 = ??$

Note that $u_1 \cdot u_2 = 0$

Poll Everywhere

Group

2 min

$$z_{i,1} = \sum_{j=1}^{5} u_1[j]\cdot x_i[j] = -\frac{1}{2}\cdot 0 + 0\cdot(-7) + \frac{1}{2}\cdot 3 - \frac{1}{2}\cdot 2 + \frac{1}{2}\cdot 5 = 3$$

$$z_{i,2} = \sum_{j=1}^{5} u_2[j]\cdot x_i[j] = \frac{1}{2}\cdot 0 + 0\cdot(-7) + \frac{1}{2}\cdot 3 - \frac{1}{2}\cdot 2 - \frac{1}{2}\cdot 5 = -2$$

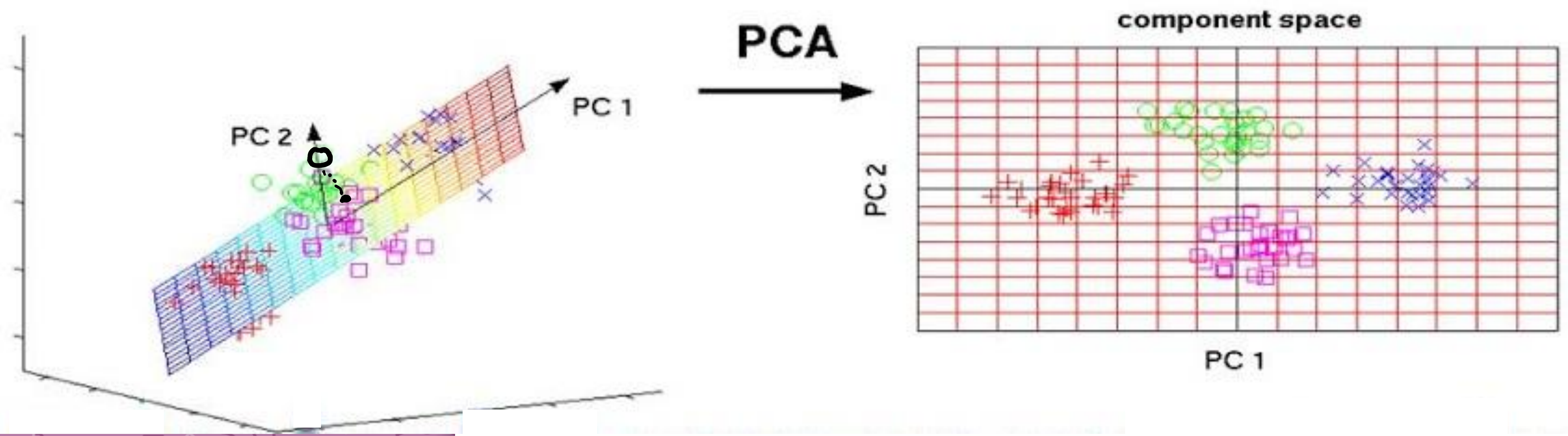
- Compute the 2D coordinates of the following point. Then compute its reconstruction error.
  - $x_i = [0, -7, 3, 2, 5]$
  - $u_1 = [-0.5, 0, 0.5, -0.5, 0.5]$
  - $u_2 = [0.5, 0, 0.5, -0.5, -0.5]$
  - $z_i =$?? $[3, -2]$
  - $\hat{x}_i =$?? $[-2.5, 0, 0.5, -0.5, 2.5]$
  - $\|\hat{x}_i - x_i\|_2^2 =$?? $(-2.5-0)^2 + (0+7)^2 + (0.5-3)^2 + (-0.5-2)^2 + (2.5-5)^2 = 74$

$$\hat{x}_i = z_{i,1}\cdot u_1 + z_{i,2}\cdot u_2$$
$$= [-\tfrac{3}{2}, 0, \tfrac{3}{2}, -\tfrac{3}{2}, \tfrac{3}{2}]$$
$$+$$
$$[-1, 0, -1, 1, 1]$$
$$= [-2.5, 0, 0.5, -0.5, 2.5]$$

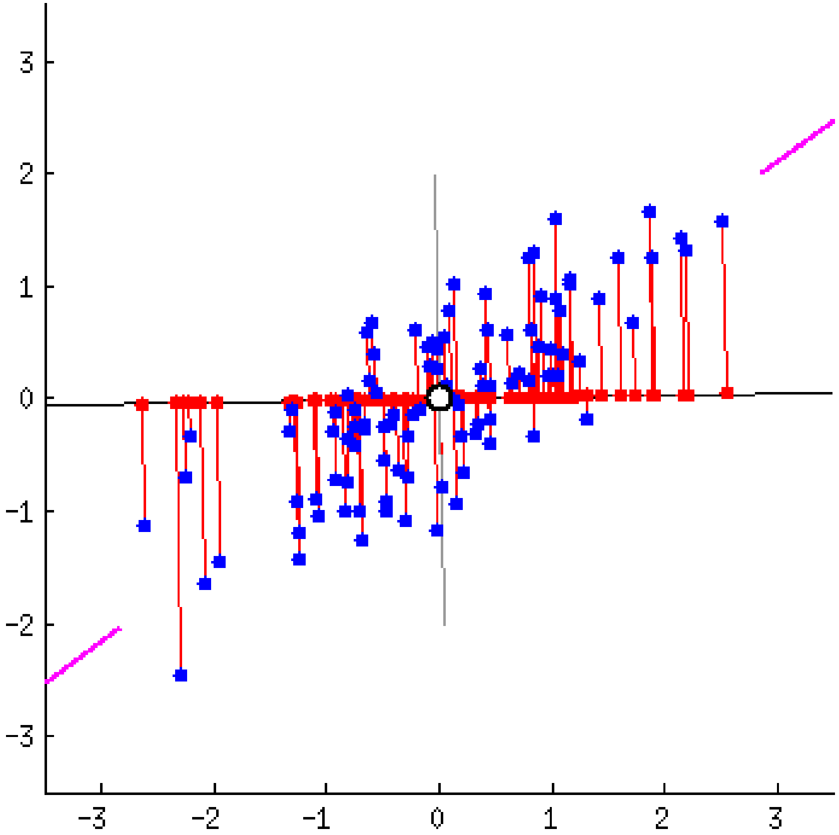
Linear Projection in Higher Dimensions

Think of PCA as giving each datapoint a new ”address.”

- Earlier, you could find the datapoint by going to the location (x, y, z).
- Now, if you are just moving in the projection plane, you can (approximately) find the datapoint by going to the location (u_1, u_2)



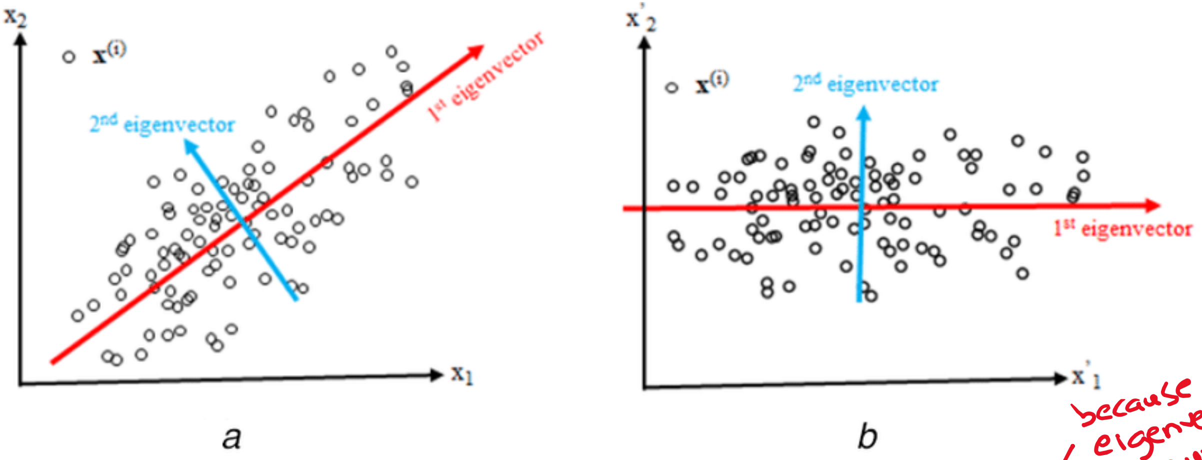
How do we find the best projection vector(s)?



Pick the vector(s) along which the datapoints have the most variation!

Eigenvectors

- There is a quantity in linear algebra that does exactly that!
- The **eigenvectors** of a d-dimensional dataset* are a collection of d *perpendicular* vectors that point in the directions of greatest variation amongst the points in the dataset.



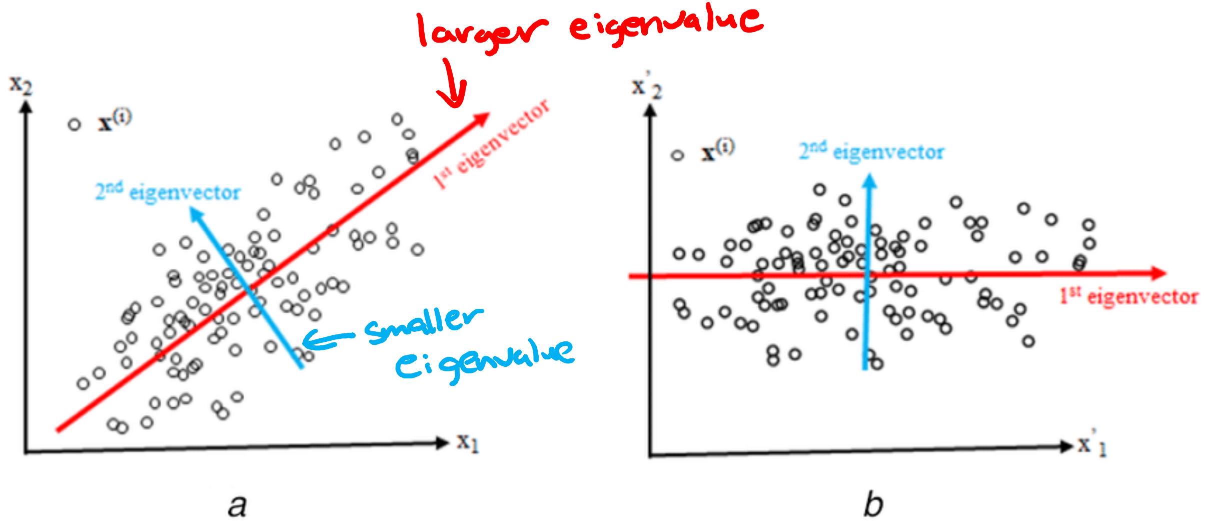
- Eigenvectors rotate the axes of the d dimensional space.

↙ because num eigenvectors is num dimensions

* (caveat) the eigenvectors are actually associated with the *covariance matrix* of the dataset

Eigenvalues

- Each eigenvector has a corresponding **eigenvalue**, indicating how much the dataset varies in that direction.
- Greater eigenvalue ➔ greater variance.



larger eigenvalue

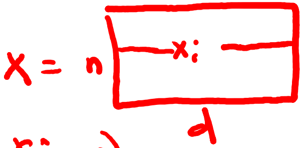
smaller eigenvalue

- **PCA**: Take the k eigenvectors with greatest eigenvalues.

PCA Algorithm

Input Data: An n × d data matrix X

- Each row is an example
- Desired k (lower dimensions)

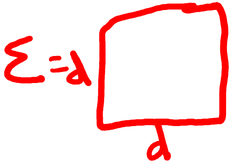


1. **Center Data**: Subtract mean from each row

$$X_c \leftarrow X - \bar{X}[1:d]$$

2. **Compute spread/orientation**: Compute covariance matrix Σ

$$\Sigma[t,s] = \frac{1}{n}\sum_{i=1}^{n} x_{c,i}[t]x_{c,i}[s]$$



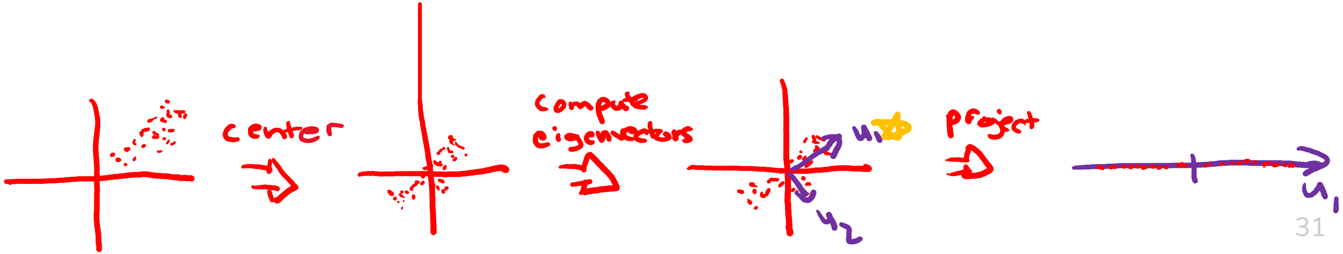
3. **Find basis for orientation**: Compute eigenvectors of Σ
   - Select k eigenvectors $u_1, \dots, u_k$ with largest eigenvalues

4. **Project Data**: Project data onto principal components

$$z_i[1] = u_1^T x_{c,i} = u_1[1]x_{c,i}[1] + \cdots + u_1[d]x_{c,i}[d]$$

…

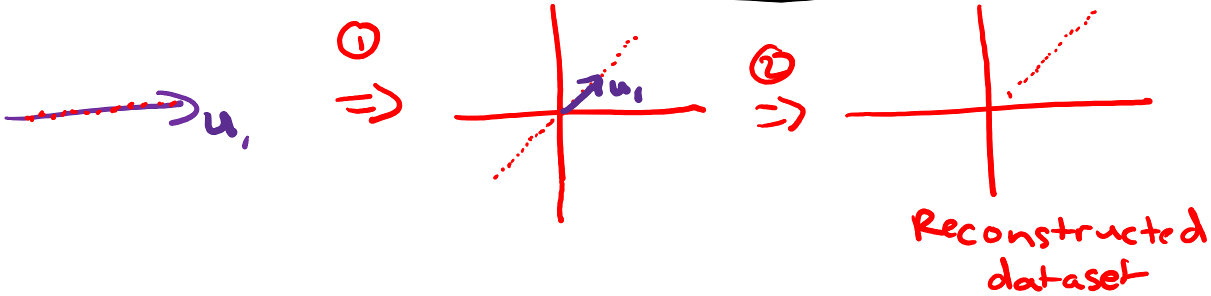
$$z_i[k] = u_k^T x_{c,i} = u_k[1]x_{c,i}[1] + \cdots + u_k[d]x_{c,i}[d]$$

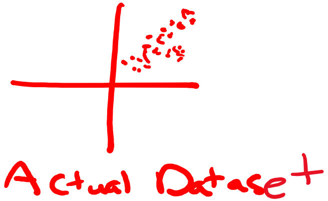


Reconstructing Data

Using principal components and the projected data, you can reconstruct the data in the original domain.

$$\hat{x}_i[1{:}d] = \underbrace{\bar{X}[1{:}d]}_{\text{add mean ②}} + \underbrace{\sum_{j=1}^{k} z_i[j]\ u_j}_{①}$$





Error for x_i

$\| \hat{x}_i - x_i \|_2^2$

Example: Eigenfaces

each image is $16 \cdot 16 = 256$

Apply PCA to face data

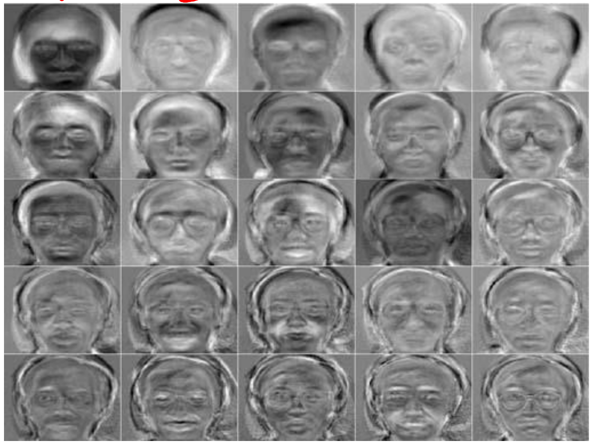
$X = 64 \cdot 256$

$z = 64 \cdot 25$

Input Data

Principal Components

u_1 u_2 u_3



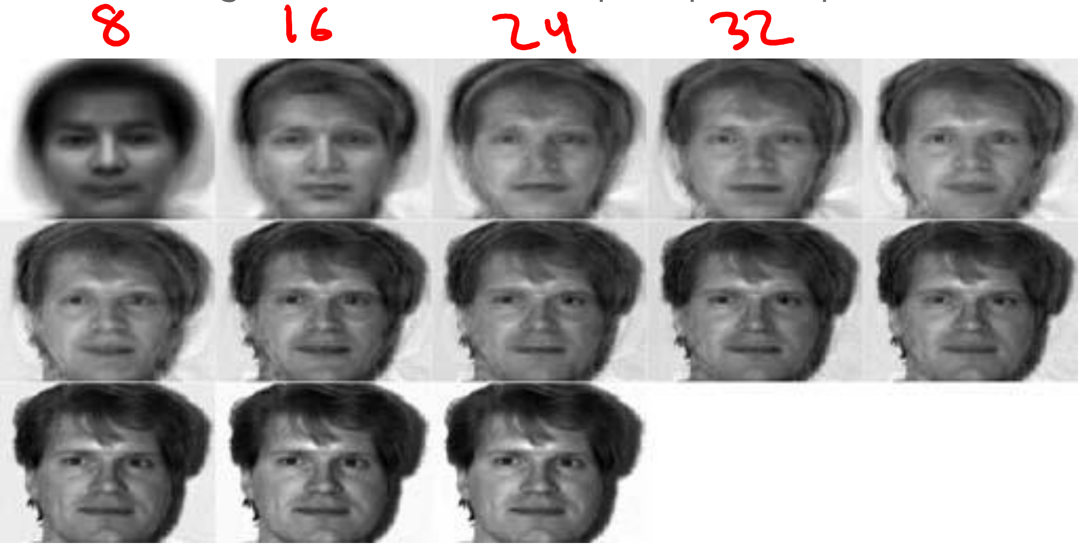
u_{25}

Reconstructing Faces

Depending on context, it may make sense to look at either original data or projected data.

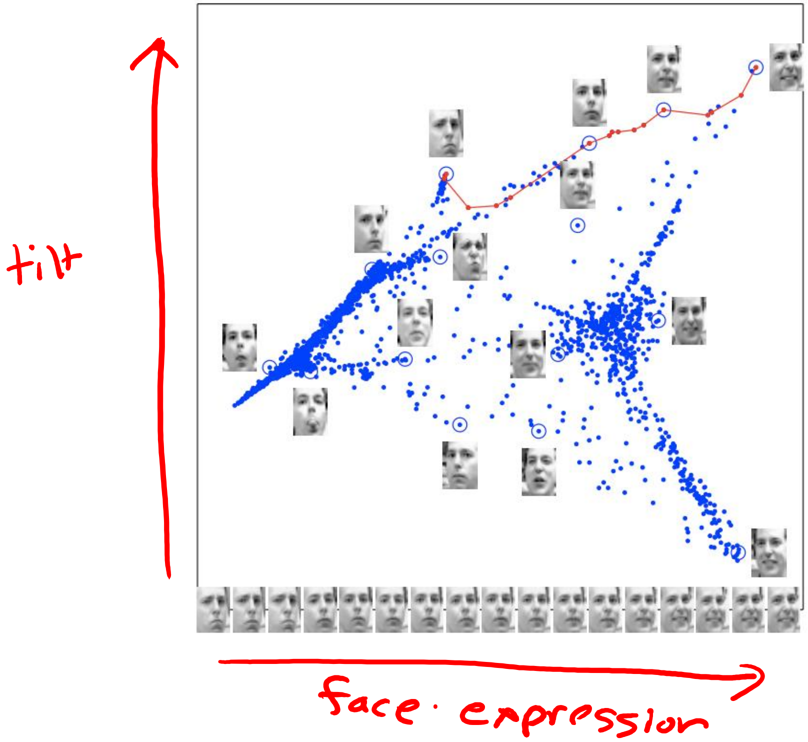
In this case, let's see how the original data looks after using more and more principal components for reconstruction.

- Each image shows additional 8 principal components



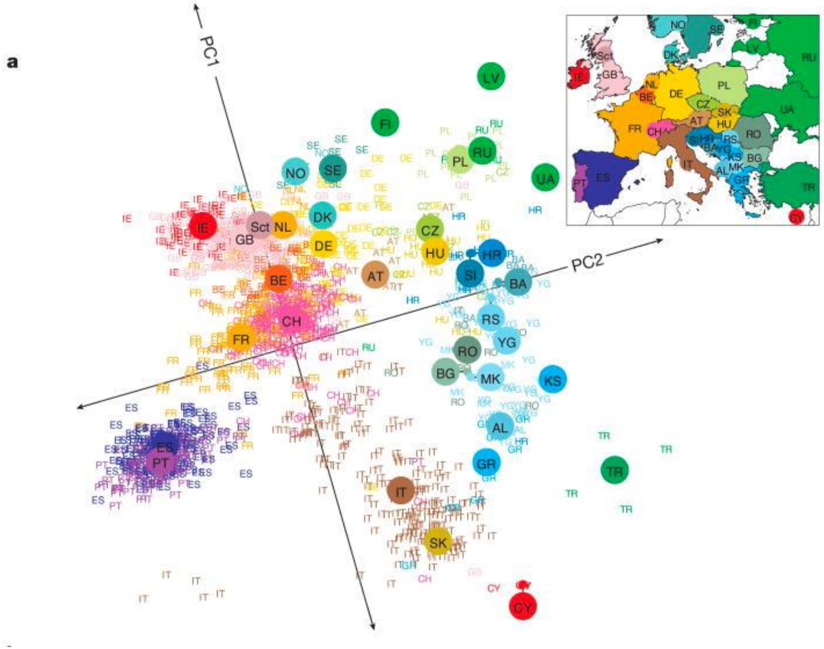
Embedding Images

Other times, it does make sense to look at the data in the projected space! (Usually if $k \leq 3$)



Example: Genes

Dataset of genes of Europeans (3192 people; 500,568 loci) and their country of origin, ran PCA on the data and plotted 2 principal components.



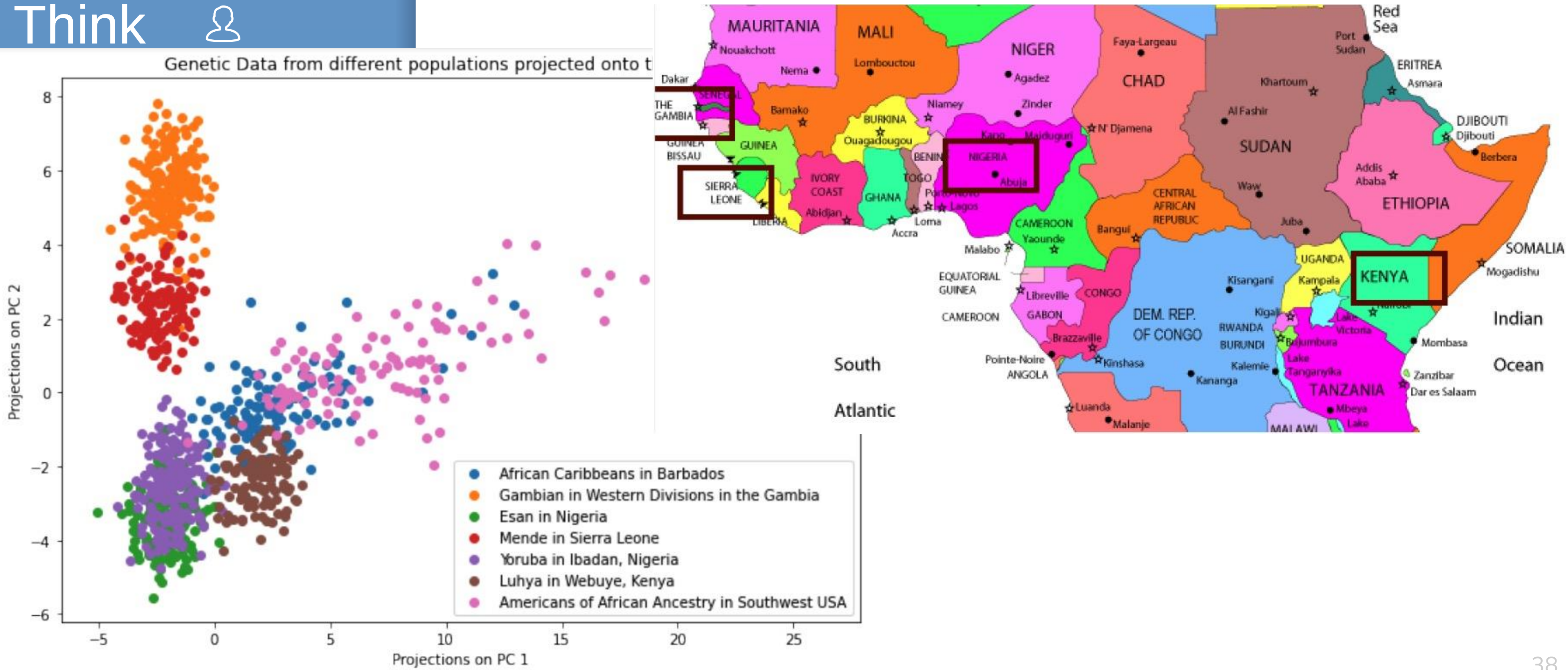
Brain Break

3:35

Poll Everywhere

Think

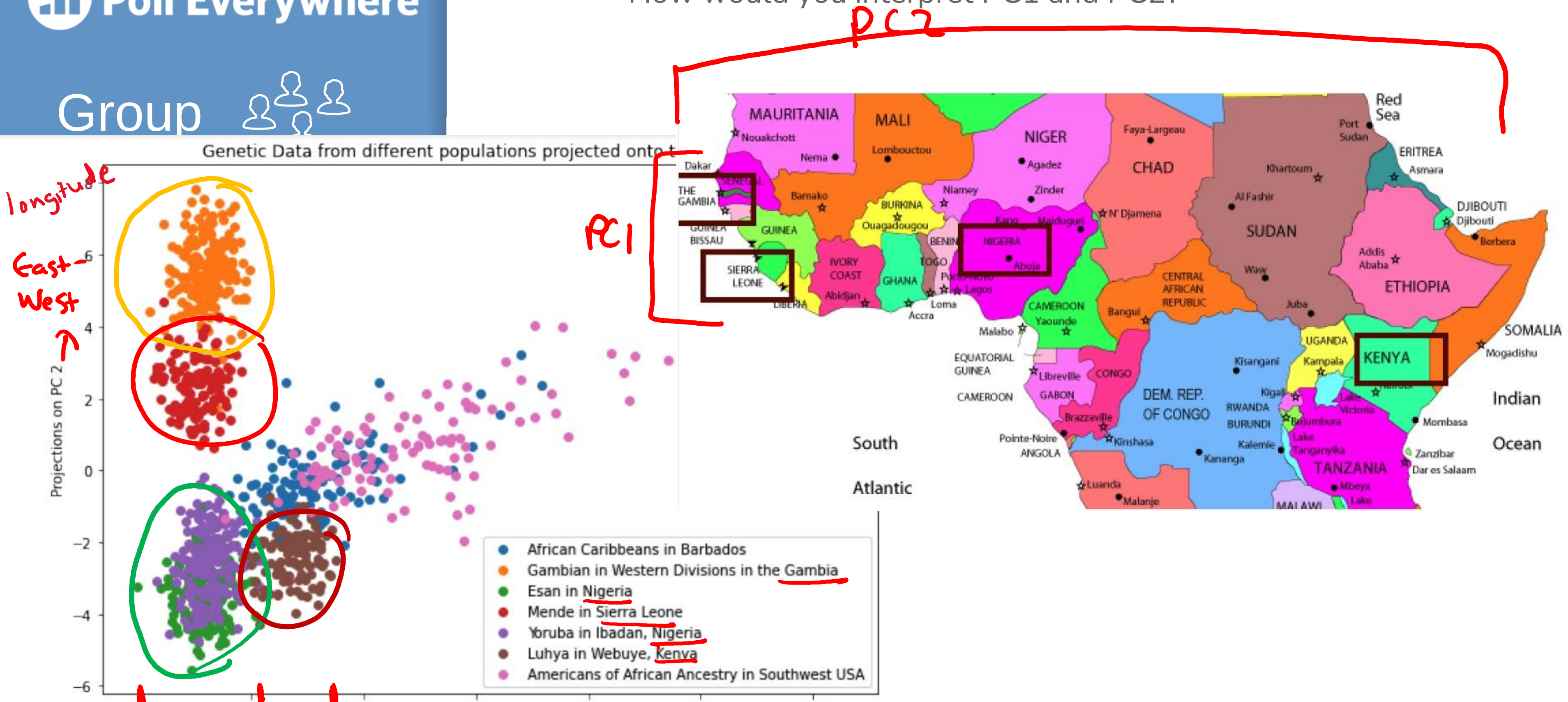
- Consider the genetic data projected onto its top two PCs. How would you interpret PC1 and PC2?



Poll Everywhere

Group

- Consider the genetic data projected onto its top two PCs. How would you interpret PC1 and PC2?



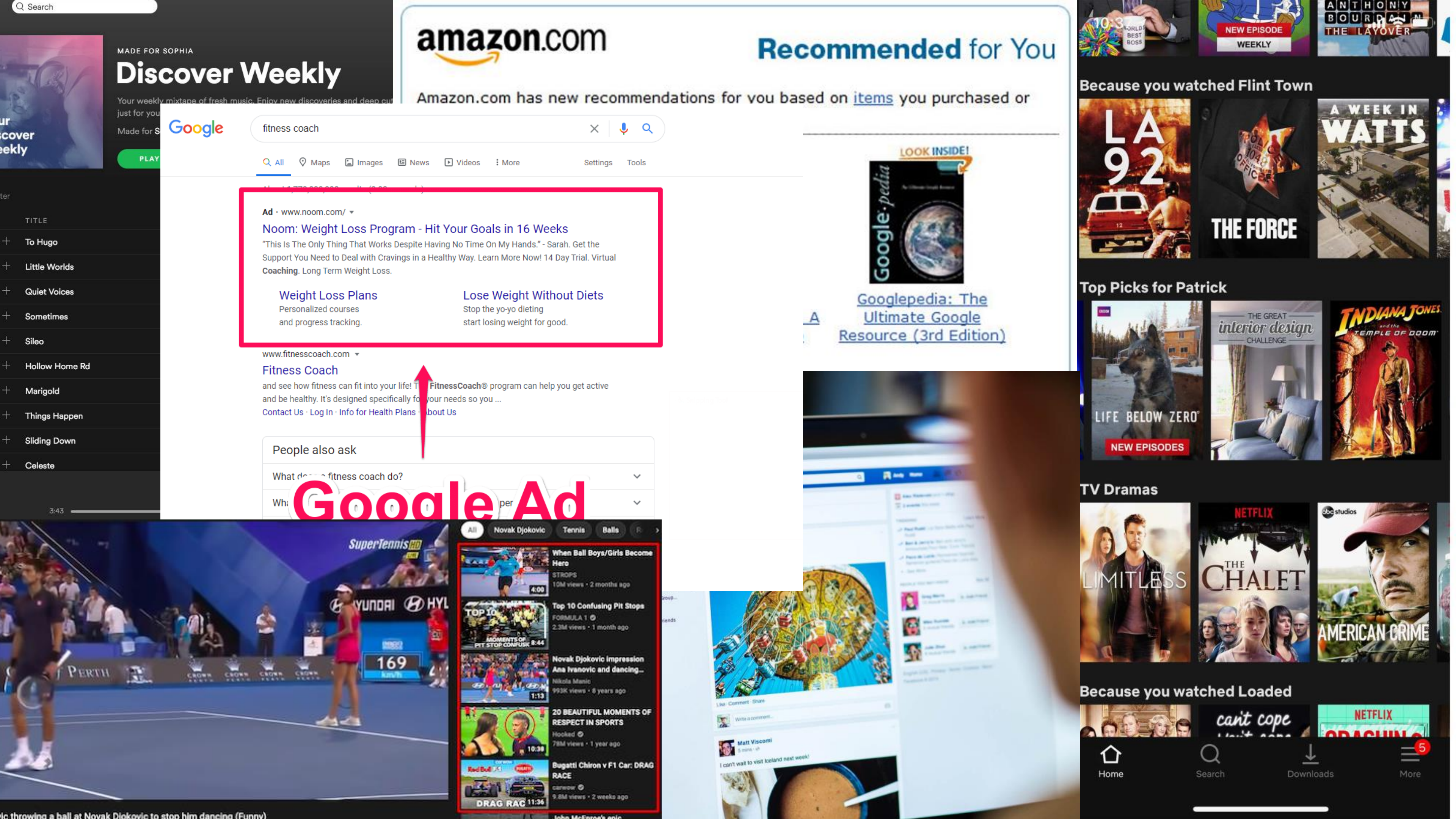
Gambia, Nigeria, Kenya, Sierra Leone → North-South, latitude

General Steps to Take as an ML Practitioner

Given a new dataset:

- Split into train and test sets.
- Understand the dataset: ←
  this is where unsupervised learning fits into a supervised learning pipeline
  - Understand the feature/label types and values
  - Visualize the data: scatterplot, boxplot, PCA, clustering
- Use that intuition to decide:
  - What features to use, and what transformations to apply to them (pre-processing).
  - What model(s) to train.
- Train the models, evaluate them using a validation set or cross-validation.
- Deploy the best model.

Intro to Recommender Systems



Search

MADE FOR SOPHIA

Discover Weekly

Your weekly mixtape of fresh music. Enjoy new discoveries and deep cut

just for you

Made for S

PLAY

TITLE

- To Hugo
- Little Worlds
- Quiet Voices
- Sometimes
- Sileo
- Hollow Home Rd
- Marigold
- Things Happen
- Sliding Down
- Celeste

3:43

amazon.com Recommended for You

Amazon.com has new recommendations for you based on items you purchased or

LOOK INSIDE!

Googlepedia: The Ultimate Google Resource (3rd Edition)

Google

fitness coach

All · Maps · Images · News · Videos · More · Settings · Tools

Ad · www.noom.com/

Noom: Weight Loss Program - Hit Your Goals in 16 Weeks

"This Is The Only Thing That Works Despite Having No Time On My Hands." - Sarah. Get the Support You Need to Deal with Cravings in a Healthy Way. Learn More Now! 14 Day Trial. Virtual **Coaching**. Long Term Weight Loss.

Weight Loss Plans
Personalized courses
and progress tracking.

Lose Weight Without Diets
Stop the yo-yo dieting
start losing weight for good.

www.fitnesscoach.com

Fitness Coach

and see how fitness can fit into your life! T **FitnessCoach**® program can help you get active and be healthy. It's designed specifically fo your needs so you ...

Contact Us · Log In · Info for Health Plans · About Us

People also ask

What d fitness coach do?

Wh per

Google Ad

When Ball Boys/Girls Become Hero
STROPS
10M views · 2 months ago

Top 10 Confusing Pit Stops
FORMULA 1
2.3M views · 1 month ago

Novak Djokovic impression Ana Ivanovic and dancing...
Nikola Manic
993K views · 8 years ago

20 BEAUTIFUL MOMENTS OF RESPECT IN SPORTS
Hooked
78M views · 1 year ago

Bugatti Chiron v F1 Car: DRAG RACE
carwow
9.8M views · 2 weeks ago

John McEnroe's epic

...ds throwing a ball at Novak Djokovic to stop him dancing (Funny)

Because you watched Flint Town

Top Picks for Patrick

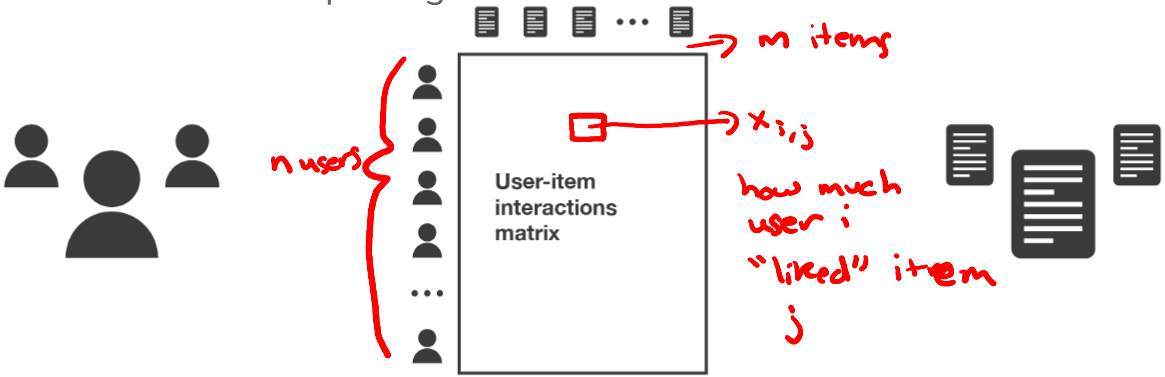
LIFE BELOW ZERO
NEW EPISODES
THE GREAT interior design CHALLENGE
INDIANA JONES and the TEMPLE OF DOOM

TV Dramas

Because you watched Loaded

Recommender Systems Setup

- You have n users and m items in your system
  - Typically, $n \gg m$. E.g., Youtube: 2.6B users, 800M videos
- Based on the content, we have a way of measuring user preference.
- This data is put together into a **user-item interaction matrix**.



Users	User-item interactions matrix	Items
suscribers	rating given by a user to a movie (integer)	movies
readers	time spent by a reader on an article (float)	articles
buyers	product clicked or not when suggested (boolean)	products
	...	

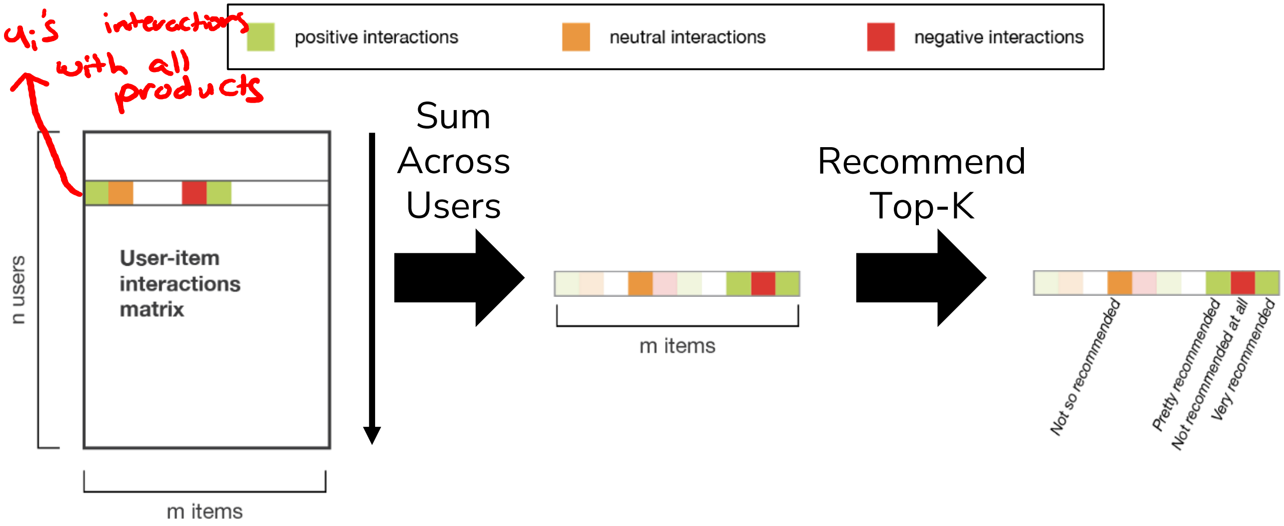
- **Task**: Given a user u_i or item v_j, predict one or more items to recommend.

Solution 0: Popularity

Solution 0: Popularity

Simplest Approach: Recommend whatever is popular

- Rank by global popularity (i.e., Squid Game)



🟩 positive interactions 🟧 neutral interactions 🟥 negative interactions

u_i's interactions with all products

n users

User-item interactions matrix

m items

Sum Across Users

m items

Recommend Top-K

Not so recommended

Pretty recommended

Not recommended at all

Very recommended

Solution 0 (Popularity) Pros / Cons

Pros:

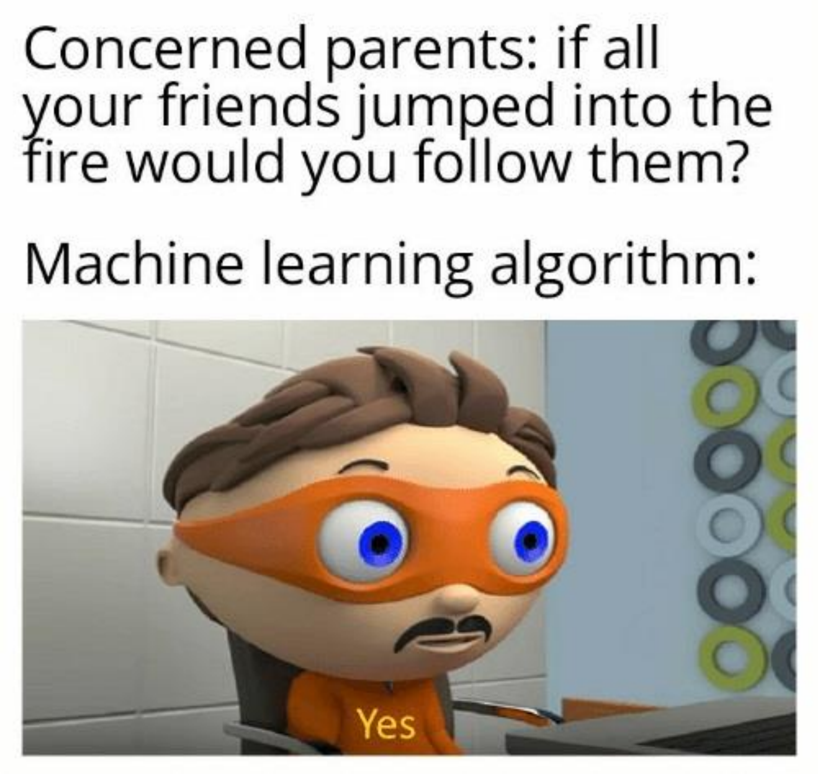
- Easy to implement

Cons:

- No Personalization
- Feedback Loops
- Top-K recommendations might be redundant
    - e.g., when a new Harry Potter movie is released, the others may also jump into top-k popularity.

Solution 1: Nearest User

User-User



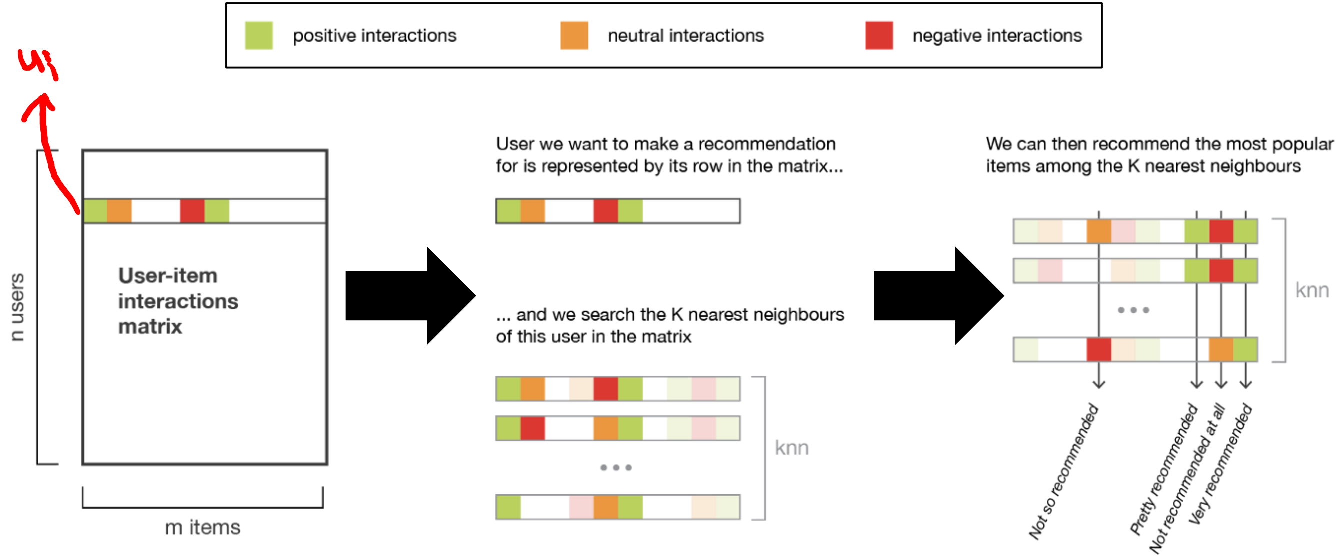
Concerned parents: if all your friends jumped into the fire would you follow them?

Machine learning algorithm:

Solution 1: Nearest User (User-User)

User-User Recommendation:

- Given a user u_i, compute their k nearest neighbors.
- Recommend the items that are most popular amongst the nearest neighbors.



🟩 positive interactions 🟧 neutral interactions 🟥 negative interactions

User we want to make a recommendation for is represented by its row in the matrix...

... and we search the K nearest neighbours of this user in the matrix

We can then recommend the most popular items among the K nearest neighbours

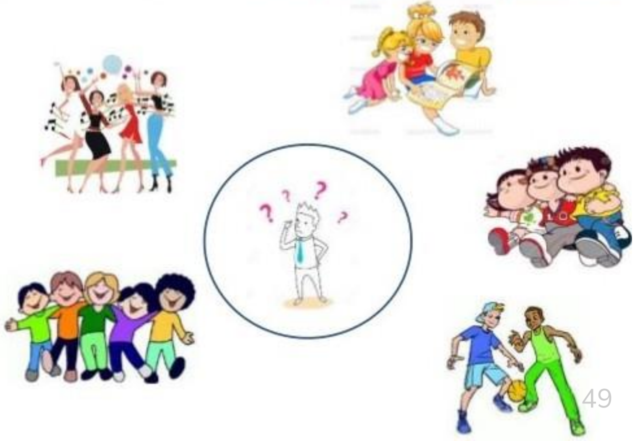
Poll Everywhere

Think

1 min

- What do you see as pros / cons of the nearest user approach to recommendations?

Tell me about your friends(*who your neighbors are*) and *I will tell you who you are*.



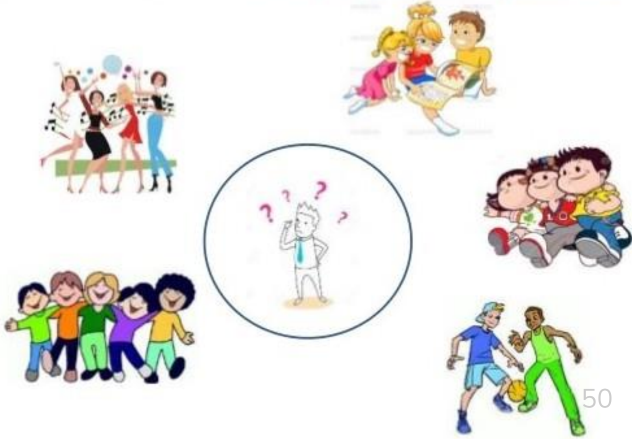
Poll Everywhere

Group

2 min

- What do you see as pros / cons of the nearest user approach to recommendations?

Tell me about your friends(*who your neighbors are*) and *I will tell you who you are*.



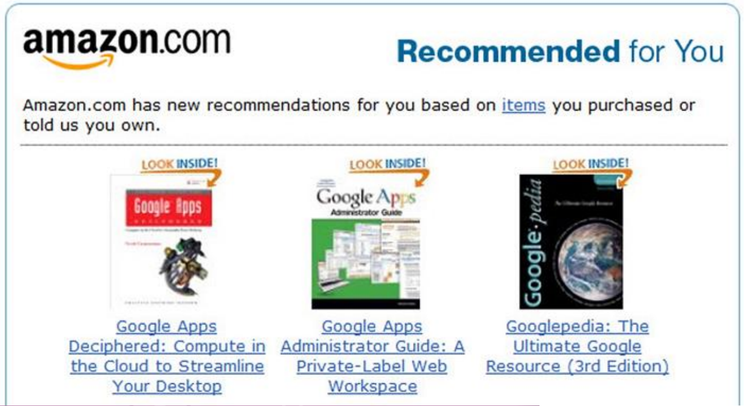
Solution 1 (User-User) Pros / Cons

Pros:

- Personalized to the user.

Cons:

- Nearest Neighbors might be too similar
  - This approach only works if the nearest neighbors have interacted with items that the user hasn't.
- Feedback Loop (Echo Chambers)
- Scalability
  - Must store and search through entire user-item matrix
- Cold-Start Problem
  - What do you do about new users, with no data?



Recommended for You

Amazon.com has new recommendations for you based on items you purchased or told us you own.

Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop

Google Apps Administrator Guide: A Private-Label Web Workspace

Googlepedia: The Ultimate Google Resource (3rd Edition)

Solution 2: “People Who Bought This Also Bought…”

Item-Item

C_{ii} = total # users who bought item i

Solution 2: “People Who Bought This Also Bought…” (Item-Item)

Item-Item Recommendation:

- Create a **co-occurrence matrix** $C \in \mathbb{R}^{m \times m}$ (m is the number of items). C_{ij} = # of users who bought both item i and j.
- For item i, predict the top-k items that are bought together.

m

m

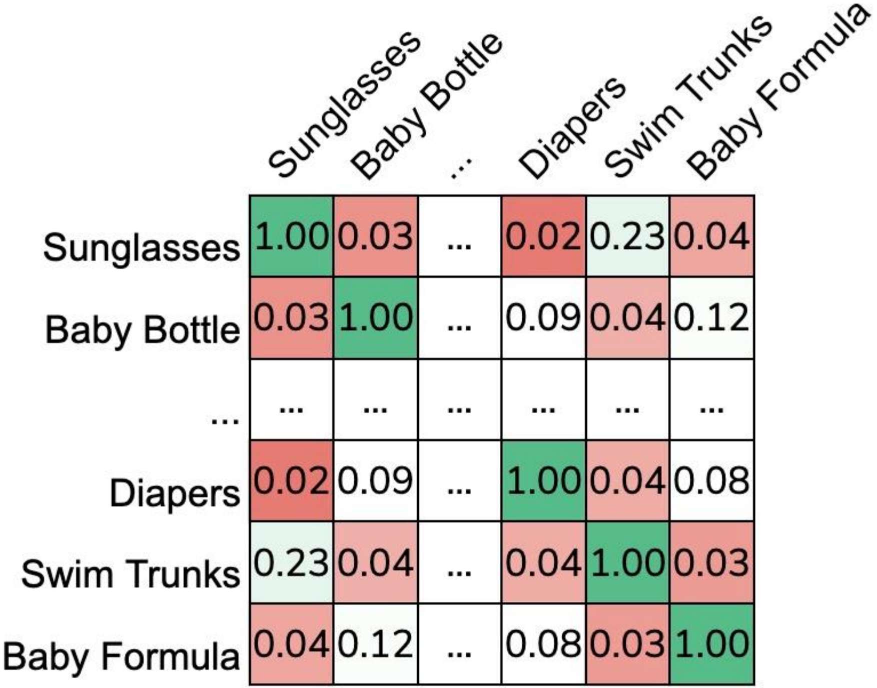
	Sunglasses	Baby Bottle	...	Diapers	Swim Trunks	Baby Formula
Sunglasses	500	15	...	9	130	20
Baby Bottle	15	45	...	6	10	10
...	...	...	...	...	...	...
Diapers	9	6	...	30	9	6
Swim Trunks	130	10	...	9	200	8
Baby Formula	20	10	...	6	8	50

Normalizing Co-Occurence Matrices

Problem: popular items drown out the rest!

Solution: Normalizing using Jaccard Similarity.

$$S_{ij} = \frac{\#\ \text{purchased}\ i\ \textbf{and}\ j}{\#\ \text{purchased}\ i\ \textbf{or}\ j} = \frac{C_{ij}}{C_{ii} + C_{jj} - C_{ij}}$$



	Sunglasses	Baby Bottle	...	Diapers	Swim Trunks	Baby Formula
Sunglasses	1.00	0.03	...	0.02	0.23	0.04
Baby Bottle	0.03	1.00	...	0.09	0.04	0.12
...	...	...	...	...	...	...
Diapers	0.02	0.09	...	1.00	0.04	0.08
Swim Trunks	0.23	0.04	...	0.04	1.00	0.03
Baby Formula	0.04	0.12	...	0.08	0.03	1.00

Incorporating Purchase History

What if I know the user u has bought a baby bottle and formula?

Idea: Take the average similarity between items they have bought

$$Score(u, diapers) = \frac{S_{diapers,baby\ bottle} + S_{diapers,baby\ formula}}{2}$$

Could also weight them differently based on recency of purchase!

Then all we need to do is find the item with the highest average score!

Poll Everywhere

Think

1 min

- What do you see as pros / cons of the item-item approach to recommendations?

Poll Everywhere

Group

2 min

- What do you see as pros / cons of the item-item approach to recommendations?

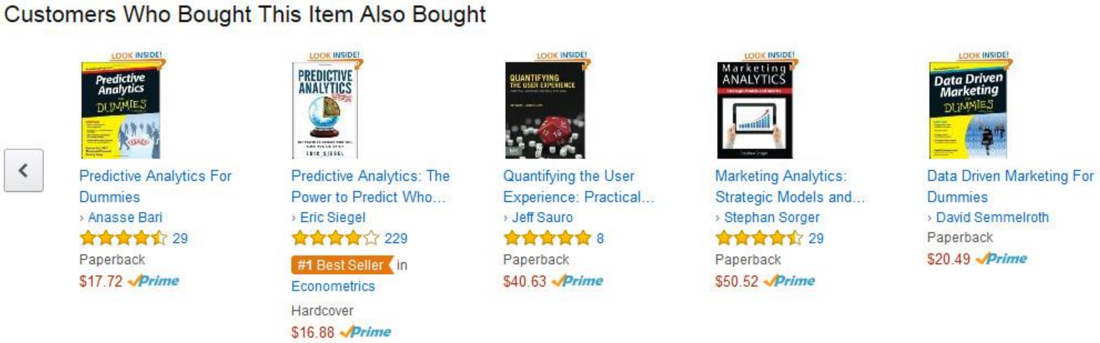
Solution 2 (Item-Item) Pros / Cons

Pros:

- Personalizes to item (incorporating purchase history also personalizes to the user)

Cons:

- Can still suffer from feedback loops
  - (As can all recommender systems – but in some cases it's worse than others)
- Scalability (must store entire item-item matrix)
- Cold-Start Problem
  - What do you do about new ***items***, with no data?



Customers Who Bought This Item Also Bought

Predictive Analytics For Dummies
› Anasse Bari
29
Paperback
$17.72 Prime

Predictive Analytics: The Power to Predict Who...
› Eric Siegel
229
#1 Best Seller in Econometrics
Hardcover
$16.88 Prime

Quantifying the User Experience: Practical...
› Jeff Sauro
8
Paperback
$40.63 Prime

Marketing Analytics: Strategic Models and...
› Stephan Sorger
29
Paperback
$50.52 Prime

Data Driven Marketing For Dummies
› David Semmelroth
Paperback
$20.49 Prime

Will pick up here on Wed

Solution 3: Feature-Based

Solution 3: Feature-Based

What if we know what factors lead users to like an item?

Idea: Create a feature vector for each item. Learn a regression model!

Genre	**Year**	**Director**	**...**
Action	1994	Quentin Tarantino	...
Sci-Fi	1977	George Lucas	...

Define weights on these features for **all users** (global)

$$w_G \in \mathbb{R}^d$$

Fit linear model

Solution 3: Feature-Based

What if we know what factors lead users to like an item?

Idea: Create a feature vector for each item. Learn a regression model!

Genre	**Year**	**Director**	**...**
Action	1994	Quentin Tarantino	...
Sci-Fi	1977	George Lucas	...

Define weights on these features for **all users** (global)

$$w_G \in \mathbb{R}^d$$

Fit linear model

$$\hat{r}_{uv} = w_G^T h(v) = \sum_i w_{G,i}\, h_i(v)$$

$$\hat{w}_G = argmin_w \frac{1}{\#\, ratings} \sum_{u,v:r_{uv}\neq ?} \left(w_G^T h(v) - r_{uv}\right)^2 + \lambda \|w_G\|$$

Personalization: Option A

Add user-specific features to the feature vector!

Genre	**Year**	**Director**	**…**	**Gender**	**Age**	**…**
Action	1994	Quentin Tarantino	…	F	25	…
Sci-Fi	1977	George Lucas	…	M	42	…

Personalization: Option B

Include a user-specified deviation from the global model.

$$\hat{r}_{uv} = (\widehat{w}_G + \widehat{w}_u)^T h(v)$$

Start a new user at $\widehat{w}_u = 0$, update over time.

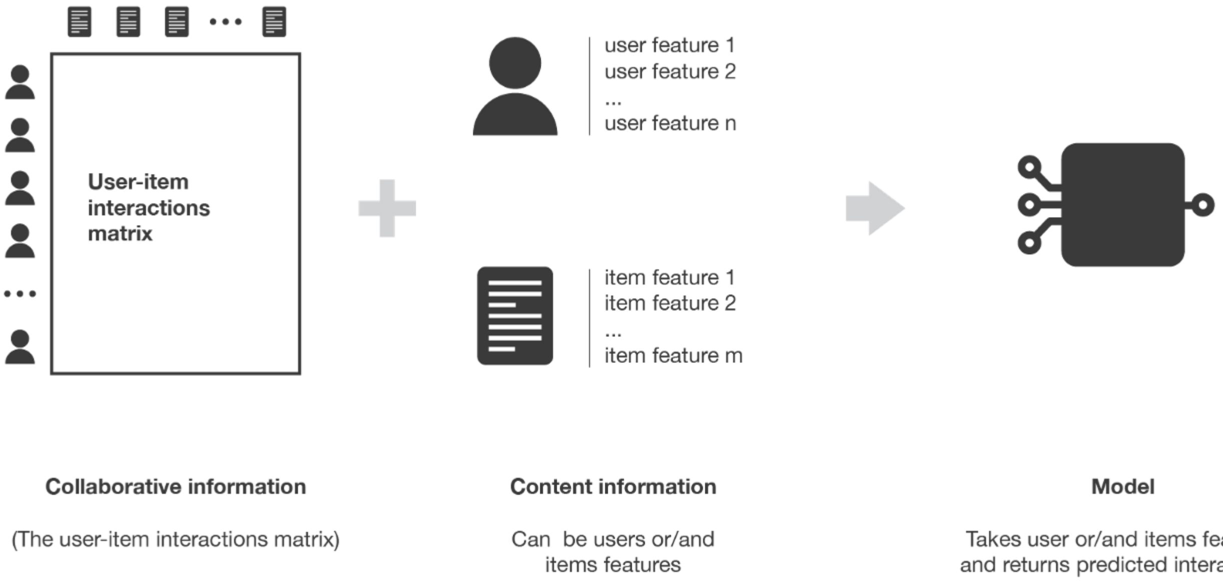
- OLS on the residuals of the global model
- Bayesian Update (start with a probability distribution over user-specific deviations, update as you get more data)

Poll Everywhere

Think

1 min

- Will feature-based recommender systems suffer from the cold start problem? Why or why not?
- What about other pros/cons of feature-based?



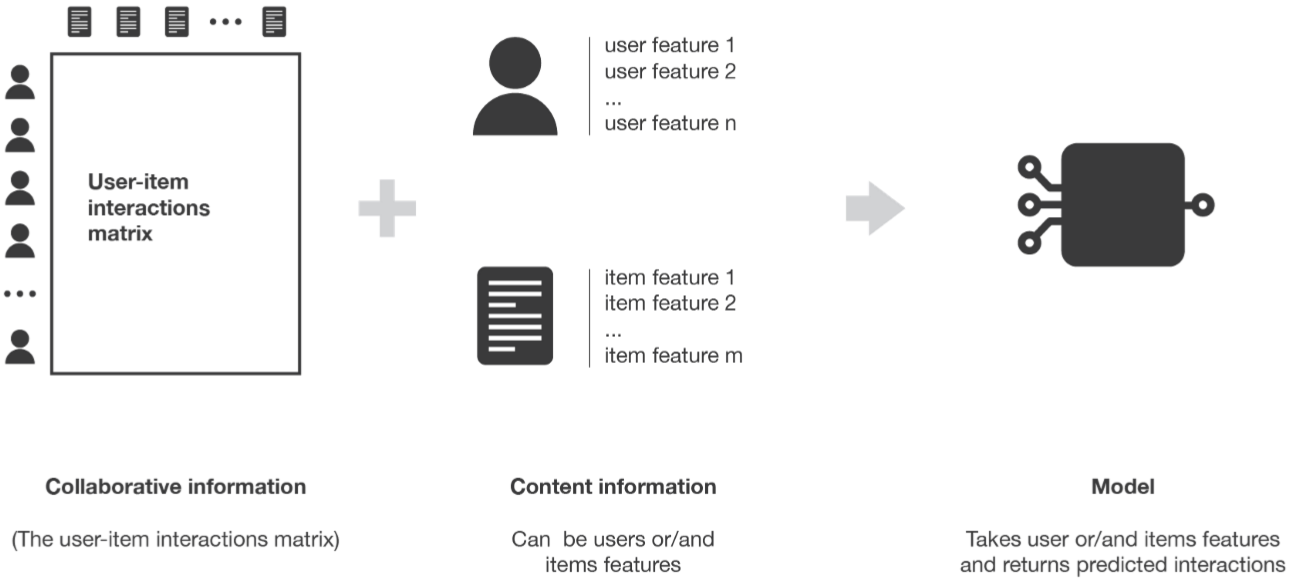
User-item interactions matrix
user feature 1
user feature 2
...
user feature n
item feature 1
item feature 2
...
item feature m
Collaborative information
(The user-item interactions matrix)
Content information
Can be users or/and items features
Model
Takes user or/and items features and returns predicted interactions

Poll Everywhere

Group

2 min

- Will feature-based recommender systems suffer from the cold start problem? Why or why not?
- What about other pros/cons of feature-based?



Collaborative information

(The user-item interactions matrix)

Content information

Can be users or/and items features

Model

Takes user or/and items features and returns predicted interactions

Solution 3 (Feature-Based) Pros / Cons

Pros:

- No cold-start issue!
  - Even if a new user/item has no purchase history, you know features about them.
- Personalizes to the user and item.
- Scalable (only need to store weights per feature)
- Can add arbitrary features (e.g., time of day)

Cons:

- Hand-crafting features is very tedious and unscalable ☹

Recap

Dimensionality Reduction & PCA:

- Why and when it’s important
- High level intuition for PCA
- Linear Projections & Reconstruction
- Eigenvectors / Eigenvalues

Recommender Systems:

- Sol 0: Popularity
- Sol 1: Nearest User (User-User)
- Sol 2: “People who bought this also bought” (item-item)
- Sol 3: Feature-Base

Next Time (Rec System Continued):

- Sol 4: Matrix Factorization
- Sol 5: Hybrid Model
- Addressing common issues with Recommender Systems
- Evaluating Recommender Systems