Lecture 13: PCA / Recommender Systems Intro

Created	@August 9, 2024 10:50 PM

Recommender System Intro

- problems
    - interests changer over time
    - you don't want recommendations to be very close, e.g. recommending all 5 movies of Dune
    - cold start makes it harder because we don't know anything about the user's interests or if it is a product, we don't know anything about how well it will do
    - scalability: lots of computations necessary
        - you can write more efficient code or approximate the solution
- models
    - popularity: not personalized, everyone sees the same thing, creates a positive feedback loop
    - classifier: takes in a bunch of input features to predict whether a user will like it or not

Dimensionality Reduction

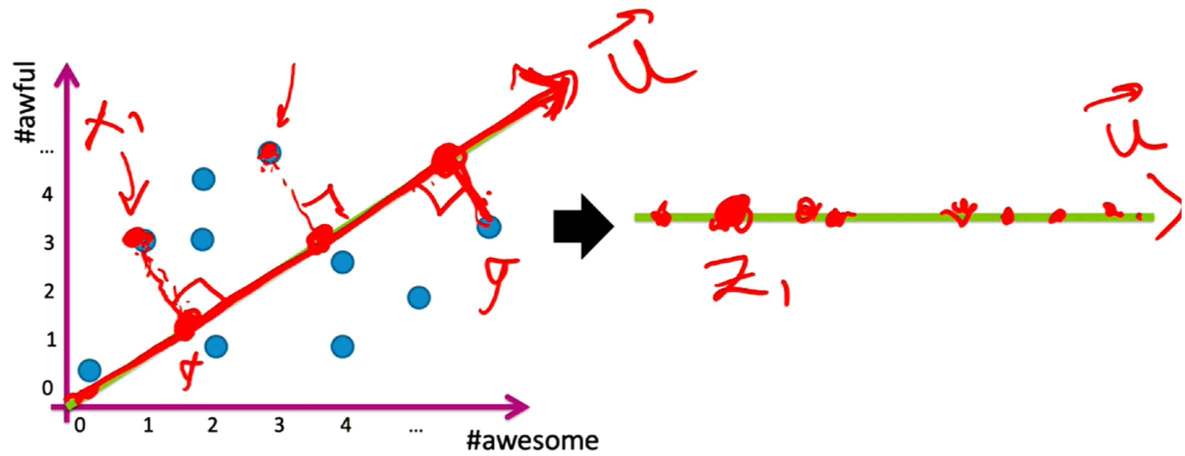
- data has a lot of dimensions
    - image: a 200×200 image has 12000 features bc rgb

- text: # of n-grams
- ratings: one rating for each object
- course success problem: many features

- why is having too many dimensions a problem?
  - hard to visualize dimensions beyond 3D
  - curse of dimensionality for KNN makes it computationally more intensive
  - overfitting/risk of too much complexity
  - much of the data is redundant/are repeats
- the goal of dimensionality reduction is to reduce the noise in the data while retaining enough signals such that a model is still able to identify meaningful relations between the data
- examples
  - embedding images in 2d
    - which direction a person is facing + their reaction (stern → happy)
  - embedding words in 1D
    - using small number of features still makes it possible to cluster meaningfully

Principal Component Analysis (PCA)

- linearly project a higher dimensional dataset to a lower dimensional one while minimizing the reconstruction error (the error that is created when mapping the lower dimensional dataset back into a higher dimensional dataset)
- linear projection: drop down each point perpendicularly onto the line
  - Z_i=u*x_i
  -



- Reconstruction: use only the projection to recreate the original dataset, with some information lost
- 3D data
  - same thing but you plot clusters in 2D
- instead of finding a data point with 3 coordinates, you can now use just 2
- the best project error is the best line of fit that point in the direction with greatest variance
  - you can use 2 eigenvectors to achieve this
  - a higher eigenvalue means that an eigenvector will have a greater variance on that particular axis
- The Algorithm: PCA
  - input data: n*p matrix X
  - center data: subtract the mean from each data point
  - compute spread using the covariance function
  - Select k eigenvectors with the highest eigenvalues
  - project the data onto the principal component using dot products between x and u
  - reconstruct the data and calculate the reconstruction data by subtracting xi^ from xi

- More principal components = higher quality
- example: genes
  - plotting the graphs of the 2 PCAs can sometimes be useful in clustering
- ML Practitioner:
  -

    Given a new dataset:

    - Split into train and test sets.
    - Understand the dataset:
      - 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.
- Recommender System Setup
  - usually the number of users is greater than the number of products
  - setup a user-item interaction matrix, e.g. for movies you would have user ratings