week 10 readings

wbg231

January 2023

1 Introduction

- the second reading is like, a new york times thing to motivate why we should care about this i don't think it is needed

a model for recommendation systems

- recommendation systems have two types of entities users and items
- we get a utility matrix showing how much the user interacted with each item what ever that means in this context.
- the matrix is sparse and we interpret empty elements as meaning we have no information about the users preference on that item
- the goal is to predict the blanks in the utility matrix
- in recommendation systems we do not need to predict a users ratings of all items we just need to get the top k items for that user
- the long tail is basically in online services there are so many products to recommend so it is required to learn individual recommendations for users
- there is implicit and explicit feedback to take into account when making recommendation systems

content based recommendation systems

- the main idea of content based recommendation systems is to focus on properties of items and recommend similar items to items similar tom those we know users like
- must construct each item a profile based on characteristic's tht can be used for recommendations

- for documents can remove stop words and use the reminaing words to compute tidf scores
- we can store item features in a vector. discrete data as booleans numeric data as numeric and as long as we keep the coloumn order consist we can take cosine similarty
- we can also make a user profile vector which has the atributes that the user interacted with
- with both a user and item vector we can use cosine similarty to recomend items
- could also view the problem in terms of building an ml classfier mdoel

colaberative filtering

- in colaberative filtering instead of looking at similarty of iterms we focous on teh similary of the users ratings for two items
- doing this requires a good distance metric
- but once we have this we can just predict any missing rating by taking the average of the n users nearest the user we are trying to rpedict for who have repdicted this item

dimensionality reduction

- another approach to recommendation systems is to view the utility matrix as the product of some user vector adn some ratings vector and try to learn thsi decompison
- bassically we just set estimate the missing valuesby minimze mean square derorr versus what we arleady know