Will keep updating this doc.

Share the thoughts of how we can approach the problem for Recommendation Feedback Project Proposal

Notes on project 535

<put name>Recommendation system 1

What are the assumptions of this system (model) and data?

What are the predictor variables and outcome variables?

What are the mathematical steps in the algorithm?

Interpretation of model

Key properties of the model

Differences with respect to other model (can be written in the end based on statistical and computing issues/differences accompanying different models)

| Movie | Alice | Bob | Carol | Dave | X\_1 (romance) | X\_2 (action) |
| --- | --- | --- | --- | --- | --- | --- |
| Love at Last | 5 | 5 | 0 | 0 | 0.9 | 0 |
| Romance forever | 5 | ? | ? | 0 | 1.0 | 0.01 |
| Cute puppies of love | ? | 4 | 0 | ? | 0.99 | 0 |
| Nonstop car chases | 0 | 0 | 5 | 4 | 0.1 | 1.0 |
| Swords vs karates | 0 | 0 | 5 | ? | 0 | 0.9 |

Our objective is to populate the ? with ratings that the users are likely to give for the movies that they did not watch.

n\_u = Number of users

n\_m = Number of movies

r(i,j) = 1 if user j has rated movie i

y(i,j) = rating given by user j movie i (defined only if r(i,j) = 1)

m\_j = number of movies rated by user j

X\_0 is the feature corresponding to the intercept term

X\_1 is the feature used to capture whether the movie is romance

X\_2 is the feature used to capture whether the movie is action

So the feature vector x\_i corresponding to the movies would look like:

x\_1 = [1, 0.9, 0], x\_2 = [1, 1.0, 0.01], x\_3 = [1, 0.99, 0], x\_4 = [1, 0.1, 1.0], and x\_5 = [1, 0, 0.9]

Regression problem can be defined as:

For each user j, learn a parameter θ^(j) belongs to R^(n+1), where n is the number of features except the feature corresponding to the intercept term. Predict user j as rating movie i with (θ^(j))^T (inner product) x^(i) stars. (θ^(j))^T is the transpose of (θ^(j)).

Suppose θ^(1) is [0,5,0]. Since x\_3 = [1, 0.99, 0], (θ^(1))^T (inner product) x\_3 = 5 \* 0.99 = 4.95, which is the rating that Alice would give to Cute puppies of Love. The objective then is to find θ^(j), i.e., parameter vector.

To learn θ^(j):

min []

θ^(j)

Repeat for every recommendation system.

What is the threshold rating for recommendation? Or rank movies in order and recommend the top rank movies first

(<https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48>)

Work with these set of inputs to algorithm to

(users, genre, movieid, rating)

(genre, movieid, rating),

(genre, movieid, gender, rating),

(age\_group, genre, movieid, rating),

(age-group, gender, genre, movieid, rating)

Approaches:

Collaborative filtering technique (<https://en.wikipedia.org/wiki/Collaborative_filtering>)

( <https://developers.google.com/machine-learning/recommendation/collaborative/basics> )

Recommends the type of movies that the active user has seen more on a regular basis or rated high for certain genre of movies by comparing these behaviors with other users of similar interests.

We can use similarity measures to recommend to users the movie they have never seen before by using the data of the users who have similar interests to that of the former.

(There is also an other technique called content based filtering which uses the properties of the product )

We can similarity measure using the following

1. Jaccard similarity---problem with this similarity is it ignores the rating values and takes into account only the movies watched in common as the similarity index
2. Cosine similarity

Centered cosine similarity ( Pearson correlation)

Can we use Association rule mining to predict movie ratings or recommend movies????

Association rule:

We can answer a question like if a set of users have watched a movie X and Y, and if the active user has watched the movie X then we can recommend him movie Y

metrics that help us understand the strength of association by

Support and confidence

( <https://towardsdatascience.com/association-rules-2-aa9a77241654> )