Recommender Systems

Sixtus Dakurah

Outline

Outline

Content-Based Recommender Systems

(1) Setup
- Given a Bucket of items **X**
- Create an **Item Profile**
  - Profile is a vector of features for a given item
  - Example: Movie {genre, actors, year, etc}
- A User takes an item from the bucket **X**
- Create a **User Profile** based on the selected items
  - Usually a vector with the same length as the item profile

(2) Goal:
- Recommend Items whose vectors are most similar to the User vector

Content-Based Recommender Systems - Example

- Consider this Items (M) profile

	Genre 1	**Genre 2**	**Actor A**	**Actor B**	**Rating**
Movie 1	0	1	1	0	2
Movie 2	1	0	1	1	4

- Also consider this Users profile (U)[1]

	Genre 1	**Genre 2**	**Actor A**	**Actor B**	**Rating**
User 1	0.5	0.001	0.3	0	2
User 2	0.2	0	1	0	1

- To recommend an item given a user profile u, we recommend the one with the highest similarity measure[2] $S(u, m)$

[1] This could be an aggregate of items the user has rated

[2] Cosine Similarity, PCC, Jaccard Similarity

Content-Based Recommender Systems - Pros & Cons

Pros

- No cold-start or sparsity problems
- Provide added information of recommended items
- Able to recommend new and unpopular items

Cons

- Feature engineering could be difficult
- Recommendation for new users can get off to a slow start
- Never recommends items outside of the users current profile.

Outline

Collaborative Filtering - Setup

(1) Upshot
- The main drive is incorporate the similarity in preferences between/among users
- Does not rely on item or user specific attributes

(2) Setup
- Given a set of users **U**
- A set of items **M**
- Let P be a (training)set of observed real valued preferences $p_{u,m}$[3] for some user(u)-item(m) pair

(3) Goal:
- Predict unobserved preferences
- That's, test set Q with pairs (u, m) not in P

(4) **This is generally posed as a matrix completion problem**[4]

[3] e.g. click/view count, item rating

[4] Other methods used to achieve this goal are KNN, Naive Bayes

Low Dimensional Matrix Factorization - Approaches

(1) Eigen Value Decomposition (EVD)
   - This works for symmetric matrices

(2) Singular Value Decomposition
   - Generalizes EVD to asymmetric(rectangular) matrices

- How SVD works
  - Decompose the preference matrix P into two lower dimensional matrices:

$$P_{UxM} = A\Sigma B^T$$

  - A_{UxK} is a matrix that represents the users latent features
  - B_{MxK} is a matrix that represents the items latent features
  - Σ_{KxK} is a diagonal matrix that measures the strength of association between users latent features and the items latent features[5]

[5] K is arbitrarily chosen to minimize the reconstruction error

Low Dimensional Matrix Factorization - Approaches

- To make a recommendation for a given user u:
  - Pick the user latent feature vector from A, denoted as $A_{(u,)}$
  - Choose an appropriate similarity measure S
  - Find the similarity $S(A_{(u,)}, B_{(m,)})$ for all items m in B
  - Recommend the item(s) to the user with the highest similarity measures
- Pros & Cons

Pros

  - Low space usage
  - Makes use of correlation among features
  - No explicit feature selection needed

Cons

  - SVD does not work well for very sparse preference matrix
  - **SVD does not in fact make predictions for unrated entries!**

- Note: the above method describes a User-Item collaborative filtering. Variants such as User-User and Item-Item collaborative filtering exist and follow similar procedures

Outline

Why Matrix factorization is not enough

- Consider the User-Item interaction[6] matrix

	m_1	m_2	m_3	m_4	m_5
u_1	1	1	1	0	1
u_2	0	1	1	0	0
u_3	0	1	1	1	0

- Let $S\{u, m\}$ denotes the similarity between user u and item m.
- We can easily see that $S\{u_2, u_3\}(0.8) > S\{u_1, u_2\}(0.75) > S\{u_1, u_3\}(0.5)$ using Jaccard similarity[7]

[6] Any preference matrix can be converted into an interaction matrix

[7] Cosine similarity will give the same ranking

Why Matrix factorization is not enough

- If we let P_i be the entry in a reconstructed interaction matrix using the cosine measure(I used Jaccard above because it's easy to illustrate)
- If we map the users unto a two-dimensional latent space, we get

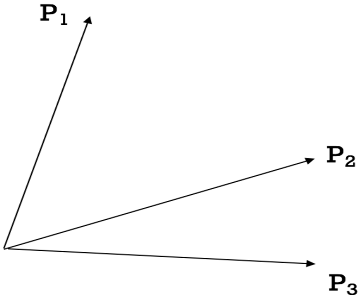

Figure: User Latent Mapping

Why Matrix factorization is not enough

- Assume we add a fourth user to the interaction

	m_1	m_2	m_3	m_4	m_5
u_1	1	1	1	0	1
u_2	0	1	1	0	0
u_3	0	1	1	1	0
u_4	1	0	1	1	1

- We observe that $S\{u_1, u_4\}(0.6) > S\{u_3, u_4\}(0.4) > S\{u_2, u_4\}(0.2)$ using Jaccard similarity

Why Matrix factorization is not enough

- The new two-dimensional user latent space mapping will now be

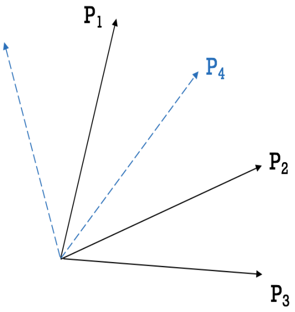

Figure: User Latent Mapping

- Placing P_4 close to P_1 keeps it far away from P_3 whiles overestimating it's closeness to P_2. Therein lies the limitation of "simple inner product"!

NCF - Setup

(1) Switching notations a little bit to match that of the paper
- Let $Y \in R^{MxN}$ be the user-item interaction matrix, with 0, 1 entries
- Let $P \in R^{MxK}$ be the user latent feature space
- Let $Q \in R^{NxK}$ be the item latent feature space[8]
- Let y_{ui} and $\hat{y}_{ui}$ denotes the observed and estimated entry entry for a given user-item interaction matrix

(2) Under matrix factorization

$$\hat{y}_{ui} = f(\mu, i | p_{\mu}, q_i) = \sum_{k=1}^{K} p_{uk} q_{ik}$$

(3) The proposed method to improve on this linear model is using MLP to introduce non-linearities

[8] Obtained via matrix factorization

NCF - General Framework

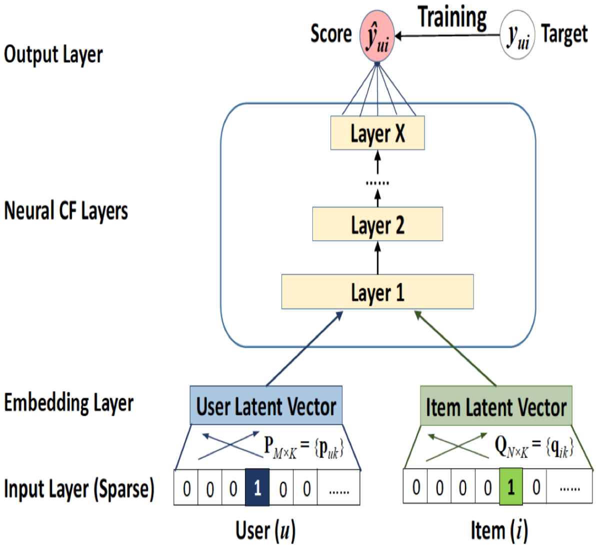

Figure: NCF Framework - Source: paper

Learning Parameters & GMF as a Special case of NCF

(1) If we define $L(y_{ui}, \hat{y}_{ui})$ to be the loss function, we can apply the usual optimization techniques to minimize this loss.

(2) It's also easy to see that Generalized Matrix Factorization is a special case of NCF. Consider the graph below:

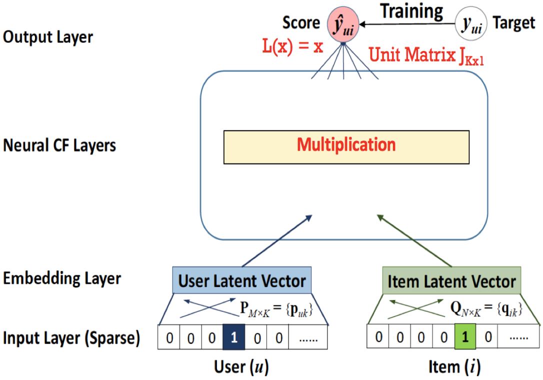

Figure: GMF as a NCF Framework

Introducing additional non-linearities

(1) In an attempt to introduce additional non-linearities, the final model proposed, NeuMF, that merges a MLP module to a GMF layer.
(2) How?
    - Train a GMF model using the frame illustrated above
    - Reconstruct the user and item latent feature space
    - Train a NCF using any combination of hidden layers
    - Also reconstruct the user and item latent feature space
    - Now concatenate[9] these four latent features to obtain a new user and item latent feature space

[9] If MLPUV and MLPIV are of shape (NxK), their concatenated matrix will be of shape (2NxK)

Introducing additional non-linearities

(1) The NeuMF framework is as illustrated below

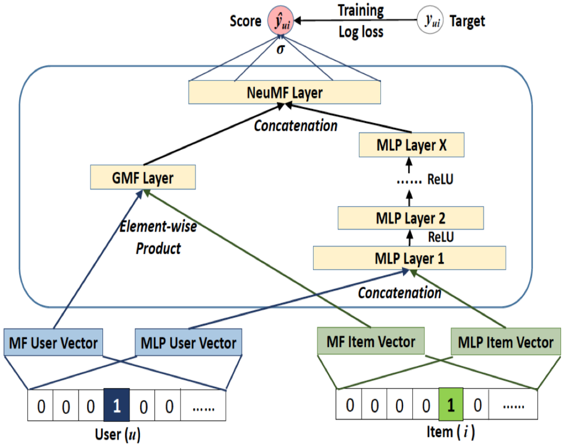

Figure: Merge of GMF and NCF

NeuMF Performance

(1) Using leave-one-out cross validation[10], the paper evaluates NeuMF and other models

(2) Two evaluation metrics were considered

- Hit Ratio @ 10
- Normalized Discounted Cumulative Gain (NDCG) @ 10

Hit Ratio @ k

- The hit ratio for a single user is determined as follows

(i) Find all rated(positive interactions) in the users history

(ii) Leave one out and feed all other items to the recommender

- Ask the recommender to recommend the top-k items
- If the left out item is in the recommended k[11], that is a hit
- To get the hit rate for the entire system, take the total hits over the number of users

[10] The last interaction of each user is held out for evaluation

[11] k is chosen taking into consideration, the sparsity of the interaction matrix