# Player Modelling Using NMF In Recommender Systems

**Ronghao Yang**
School of Computer Science
University of Waterloo
Waterloo, ON, N2L 3G1
r39yang@uwaterloo.ca

## Abstract

Player modelling methods are commonly seen in video games. Such methods are implemented to improve players' user experience. Other than being popular in video games, player modelling methods can also be used for recommender systems. Users are being modelled by such methods so that a corresponding item can be recommended to the user based on his/her user type.

## 1 Introduction

Have you ever wondered, why can't you find the best music on Spotify? Or the most interesting book on Amazon? Or the finest hotel in the city of New York? In today's world, we want the service we get from the service providers (no matter online or offline) to be tailored to our interests, which means the services these days better to be personalized to amaze the customers. This is why recommendation systems are crucial in such business applications.
For this project, I have implemented a player modelling algorithm called $NMF$ for a hotel recommendation problem.

## 2 Non-negative Matrix Factorization (NMF)

- Introduction to NMF

$NMF$ is a matrix factorization algorithm which factorize a big matrix $V$($m$ by $n$) into two smaller matrices $W$($m$ by $r$) and $H$($r$ by $n$).

$$V \approx W \times H$$

For each column $v_i$ in $V$, we have

$$v_i \approx W \times h_i$$

where $h_i$ is the corresponding column in $H$, in other words, every column in $V$ is a linear combination of $W$ where $H$ is the coefficient matrix. Geometrically, $NMF$ projects the data points in higher dimensional space to the lower dimensional space formed by the basis vectors in $W$, and $H$ contains the projected coefficients.
To integrate the theory with the context, matrices are commonly seen in recommendation problems, with columns and rows being users and the corresponding items. When $NMF$ factorizes such a matrix into $W$ and $H$, the columns in $W$ contains the hidden features of the original matrix. Each basis vector in $W$ can be viewed as basic user type, every user therefore is represented as a linear combination of such basic user typrs which are .

$$u = a_1 w_1 + a_2 w_2 + .... + a_r w_r$$

where $u$ is a single user and $a_i s$ are the coefficients. Besides, such user-item matrices are usually

sparse (with high percentage of missing values), $NMF$ with EM algorithm can reconstruct the original matrix by filling out the missing values.
Here is an example of how $NMF$ works, the number of basis was set to be 100(which might not be optimal in this case):

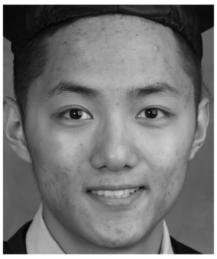

Figure 1: Original face image without any missing values

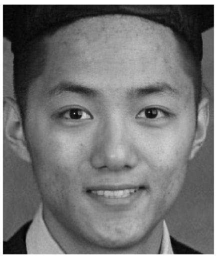

Figure 2: Face reconstruction with 50% missing values in the original image

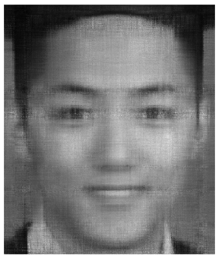

Figure 3: Face reconstruction with 90% missing values in the original image

## • Related work

In 2014, Yu & Riedl from Georgia Tech have published a paper[1] about a recent success of $NMF$ for interactive narrative recommendation system. The research was to build a drama manager that learns a model of the player's storytelling preferences and automatically recommends a narrative experience that is predicted to optimize the player's experience while conforming to the human designer's storytelling intentions [1, p. 1].
In their research, a new method called $Prefix - Based\ Collaborative\ Filtering$ (PBCF) [1, p. 2] has been introduced in which each prefix is a sequence of story plots. Based on $PBCF$, a prefix-rating matrix was constructed in which each row represents a prefix, each column represents a player, every entry in the matrix is the numerical rating rated by a player for a prefix. Similar to most recommendation problems, this matrix is sparse, due to the nature that it is impossible for a single player to encounter all the prefixes.

Figure 4: Prefix rating matrix $[1, p. 4]$

| Prefix | User 1 | User 2 | User 3 | ... |
|---|---|---|---|---|
| A (1) | * | * | 2 | ... |
| B (1, 2) | 1 | * | 2 | .... |
| C (1, 2, 6) | * | * | * | ... |
| D (1, 2, 3) | 4 | 3 | * | ... |
| ... | ... | ... | ... | ... |

$NMF$ was applied to this matrix to learn the player types so that the prefix that has the highest rating is recommended to the reader.

## • NMF algorithm

In $Algorithms\ for\ Non - negative\ Matrix\ Factorization$ published by Daniel F. Lee and H. Sebastian Seung in 2001, several $NMF$ updating rules have been introduced. One of which is

$$H_{\alpha\mu} = H_{\alpha\mu}\frac{(W^T R)_{\alpha\mu}}{(W^T WH)_{\alpha\mu}}, W_{i\alpha} = W_{i\alpha}\frac{(RH^T)_{i\alpha}}{(WHH^T)_{i\alpha}} \text{ [2, p. 3]}$$

# 3 Experiment

## 3.1 Dataset

The dataset we use for experiment is the Expedia hotel recommendation dataset from Kaggle competition in 2016. This dataset contains the hotel booking information of more than 2,000,000 users, of which the training set is obtained from 2013 and 2014 user data and the test set is obtained from 2015 user data.
In the data set, each column represent a user, each column contains a feature of the user. All of the feature variable are non-negative numerical variable except for date variable. For the purpose of simplifying of the data set, I have removed the date columns. Then the user types will only be represented by numerical values.
For privacy purpose, $Expedia$ has encoded some of the feature values, which makes the problem harder since the original values have changed. Moreover, some of the features contain missing values. This might also create some challenge to the problem.

## 3.2 User modelling and Feature selection

When using $NMF$ for building the user model, each basis user type is represented by the combination of different features. For example, assume we have $W$ as a user model which has 4 columns $(w_1, w_2, w_3, w_4)$, $w_1$ represent users who love luxurious hotels, $w_2$ represent users who prefer cheaper hotels, $w_3$ represent users who want to live in downtown, $w_4$ represent users who dersire great hotel service. Then a new user maybe of 10% of type 1, 30% of type 2, 20% of type 3 and 40% of type 3.
For feature selection, in some cases, we may also be able to select the number of basis based on some prior or domain knowledge. However, in our case, no proven knowledge is available. $NMF$ is capable of selecting the number basis user type by running cross validations. The number of basis that generates the smallest cross validation error is selected.
When running cross validation, $10-fold$ cross validation is selected. The loss metric is set to the rmse value between two matrices.

$$rmse_{A,B} = \sqrt{avg((A-B)^2)}$$

• Algorithm
For this project, $KNN$ has been implemented as a complementary algorithm to $NMF$ for predicting the hotel clusters. Each user type is simply defined by the numerical features in the training set, such as search location, etc.
In the beginning of the training process, we apply $NMF$ on the training set to compute the user model $W_{train}$ and the user coefficients of the training set$H_{train}$ . Then we apply the computed model $W_{train}$ on the testing set to obtain the coefficients ($H_{test}$) of the testing users. Once we have the coefficients of a testing users, we know what types of users they are. Then we go back to the training set and use $KNN$ to find what hotel cluster users that have the similar coefficients choose, then we use that hotel cluster as a prediction for the unknown users. The algorithms are the following:
For computing the user coefficients, we use the same algorithm [1, p.6] introduced in Yu and Ridel's paper.

**input** **:** User model $W_{train}$, Initial $R_{train}$ with missing values
**output** **:** User coefficients H

*Initialize H*;
**while** *not convergent* **do**
    Compute R' using $R' = W \times H$
    Set the corresponding number in $R'$ to be known values in $R_{train}$
    Recompute H using $H_{\alpha\mu} = H_{\alpha\mu} \frac{(W^T R)_{\alpha\mu}}{(W^T W H)_{\alpha\mu}}$
**end**

**Algorithm 1:** User Coefficients Prediction Algorithm

Once we have the user coefficients of both training set and testing set, we apply KNN(K-nearest neighbours) on $H_{train}$ and $H_{test}$ for clustering.

**input** : User coefficients in training set $H_{train}$
User coefficients in testing set $H_{test}$
Clusters for training set $C_{train}$
Number of clusters k
**output** : Predicted clusters for testing set $C_{test}$

**foreach** $h_i$ ***in*** $H_{test}$ **do**
  Find k nearest points in $H_{train}$ using KNN
  For these k nearest points, find the majority of their corresponding clusters in $C_{train}$
  Set $c_i$ in $C_{test}$ to be that cluster
**end**

**Algorithm 2:** Cluster Prediction Using KNN

# 4 Results

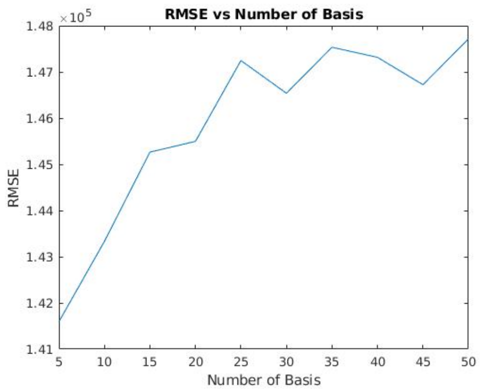


Figure 5: Cross validation with number of basis 5, 10, 15, .., 50

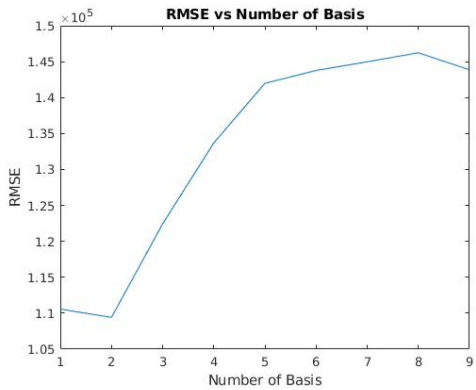


Figure 6: Cross validation with number of basis 1, 2, 3, ...9

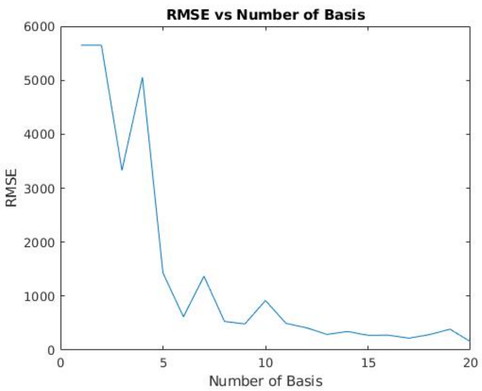


Figure 7: Testing basis without using cross validation

The table below shows the accuracy of the first method, r is the number of basis in the user model, k is the knn parameter.

| Results using method 1 | | | | | |
|---|---|---|---|---|---|
| | r=2 | r=5 | r=10 | r=15 | r=20 |
| k = 1 | 2.18% | 2.20% | 2.36% | 2.31% | 2.39% |
| k = 3 | 2.11% | 2.17% | 2.25% | 2.20% | 2.22% |
| k = 5 | 2.14% | 2.18% | 2.24% | 2.28% | 2.24% |
| k = 7 | 2.14% | 2.29% | 2.22% | 2.36% | 2.26% |
| k = 9 | 2.19% | 2.21% | 2.29% | 2.25% | 2.34% |

As we can see, the results are disappointing. There are 100 hotel clusters in the dataset, random guessing gives around 1% accuracy. The accuracy reached by NMF algorithm is just slightly better than random guessing. This phenomenon might come from different factors. Look into some details of the dataset,

## 4.1 Comparison with other algorithms

As in the papers and reports regarding the Expedia hotel recommendation competition, $NMF$ has never been implemented.

# 5 Extra Experiments

To have deeper insights of $NMF$ and analyse what factors may affect the performance of $NMF$. I have also performed several experiments on $NMF$ and applied the algorithm on a different dataset.

- Number of Basis

- Percentage of Missing Values

- Percentage of Missing Values

- NMF on Movie Dataset

# 6 Conclusion

# 7 Discussion

As we have stated earlier in the introduction to NMF section, one of the most important assumptions of $NMF$ is linearity in the dataset, which assumes that a user can be represented as a linear combination of basis user types.

# 8 Note

All the algorithms used for this project, including $NMF$, $KNN$ have been written by myself, no libraries have been used. Source code is available upon request.

## 9 Acknowledgement

I greatly acknowledge Dr.Yaoliang Yu for the amazing knowledge he shared with us and his support through out the term.

## References

[1] Hong Yu and Mark O. Riedl. *Personalized Interactive Narratives via Sequential Recommendation of Plot Points* IEEE Transactions on Computational Intelligence and AI in Games, 6(2):174–187, 2014.

[2] Daniel D. Lee and Seung, H. Sebastian. *Algorithms for Non-negative Matrix Factorization* Advances in Neural Information Processing Systems 13, 556–562, 2001

[3] Kaggle- Expedia Hotel Recommendations, *https://www.kaggle. com/c/expedia-hotel-recommendations* Accessed: 2017-10-30.