Matrix Completion through Matrix Factorization and Autoencoder
Subtitle: Matrix Completion for Students' Psychological Survey Scores

Yunfei Luo YUNFEILUO@UMASS.EDU
Amherst, MA
University of Massachusetts Amherst

Abstract

Nowadays, college students are facing multiple source of stress during the school life. There are psychological surveys that could reveal the situation of students' happiness, stress, and how they behave in their social networks. These evaluation methods reveal great clinical applications such as psychotherapy. However, there are many such surveys, it is somehow tedious for students to do all these surveys. As a result in the real-life scenario, their are missing scores from some students. This project focus on the matrix completion task. I will present the performance from conducting experiments with different methods include Matrix Factorization and Multi-layer Perceptron Autoencoder.

1. Introduction

With the advanced development of the online platform, there are many applications that are frequently used by people, such as online shopping, reading e-books, and rating movies. The actions of the customers on these platform could be stored and used for analysis purpose. Such analysis reveal great applications in business management Choi et al. (2017), customer behavioral prediction, and recommendation system Nagarnaik and Thomas (2015).

In the field of psychotherapy, the patients are often been asked to fill some surveys. There are some typical surveys, such as PHQ-9 for depression, Perceived stress scale (PSS) for stress, Longliness scale, Flourish scale, Positive and Negative Affect Schedule (PANAS), and Big Five Personality Traits. The surveys scores can be used as the features of the patients, that could help the psychologist providing recipes or behavioral suggestions Taylor et al. (2020). Such application is essentially similar with recommendation system.

Recommendation system is one of the fields that works closely with the customers actions. For example, in movie recommendation, the table that contain the rating of different movies from each customer will be used for evaluate the preference level given a pair of customer and movie item. However, in real life scenario, such table is very sparse, because it is impossible that every customer will watch all the movie. Collaborative filtering Su and Khoshgoftaar (2009) is often applied on such table to fulfill the missing entries, in order to predict the preference level. The motivation behind collaborative filtering is that such method assumes people are similar with some other people. If one customer didn't watch a movie, we can impute this missing rating by inspecting how this customer rate other movies, and finding other people who had rate the movie we are interested in and also had provided similar ratings for other movies with the customer we are looking at, then approximate the missing rating by averaging or selecting the mode over the exist ratings.

Collaborative filtering have been one of the golden choice to perform such tasks like rating or survey scores completion. As deep learning had revealed its advancement in many fields, it is also worth to adopt neural networks on these tasks Fan and Chow (2017). With such supervised machine learning method, the input is the matrix, rating table for instance, and the supervised label is the matrix itself. So the model is essentially trying to find the hidden true matrix that is closest to the matrix we have.

In this project, I experimented with both collaborative filtering and deep learning approaches. I also proposed a combined version of these two methods, and present the performance comparison.

2. Related Work

Matrix Completion has been a classical problem. The approaches to such problem reveal great applications in the fields like recommendation system. Collaborative filtering, Su and Khoshgoftaar (2009) is the well-known technique that used to conduct the task. Park et al. (2014) have tried to use K-Nearest Neighbor to implement the Collaborative filtering.

Another dominant implementation for collaborative filtering is matrix factorization Koren et al. (2009). The authors proposed a method that decompose the observed matrix to two sub-matrices. Such approach is intending to find the hidden features for each users and items.

Another classical approach is rank minimization. In this method, the objective function is the nuclear norm, which is the sum of the singular values. The constraint here is that the observed entries should always remain the same. Finally, a novel approach, that apply deep learning to solve the problem. Fan and Chow (2017) have tried to train an autoencoder to reconstruct the observed entries, then use the final output of the network as the prediction. The authors indicated that such approach could return promising result.

3. Methods

3.1. Problem Setup

The task we are focusing on is essentially solving a matrix completion problem. We have N students and M surveys. We have an ground truth matrix $X \in \mathbb{R}^{N \times M}$, and an observed matrix $X_\Omega \in \mathbb{R}^{N \times M}$ where Ω is the set of known values. We want to find some function f with parameters θ, in order to minimize

$$min_\theta ||X - f_\theta(X_\Omega)||$$

More specifically, the objective function with L2 loss function would be

$$min_\theta \sum_{i,j\in\Omega} (X_{\Omega,i,j} - f_\theta(X_\Omega)_{i,j})^2 + R(\lambda, \theta)$$

where $R((\lambda, \theta)$ is the regularization term, and λ is the constraints coefficients.

3.2. Matrix Factorization

The first approach that I'm going to try is matrix factorization, i.e. we have

$$f_\theta(X_\Omega) = PQ^T$$

where $\theta = \{P, Q\}$, $P \in \mathbb{R}^{N\times H}$ matrix that specify the embedding features for each student, and $Q \in \mathbb{R}^{M\times H}$ specify the embedding features for each survey. This method assumes a linear relationship between the features of students and surveys.

3.3. Multi-layer Perceptron Autoencoder

The second approach is a Neural Network approach, where we will have a two-layer fully connected layers that form an autoencoder. So we have

$$f_\theta(X_\Omega) = \sigma_2(W_2(\sigma_1(W_1 X_\Omega + b_1)) + b_2)$$

where σ is the activation function, W is the weight matrix, and b is the bias term. This approach contains non-linearity, because the activation function is often non-linear.

3.4. Autoencoder with Matrix Factorization as Embedding

Finally, I will use the results from matrix factorization as pre-trained embeddings for students and surveys, and train an autoencoder to see if the performance could be improved. More specifically, the objective function would be

$$min_\theta \sum_{i,j\in\Omega} (X_{\Omega,i,j} - \sigma_2(W_2(\sigma_1(W_1 PQ^T + b_1)) + b_2)_{i,j})^2 + R(\lambda, \theta)$$

Where the final reconstructed matrix is $\sigma_2(W_2(\sigma_1(W_1 PQ^T + b_1)) + b_2)$.

4. Data Preparation

I use the survey responses in the dataset StudentLife Wang et al. (2014). This dataset was collected by Dartmouth college. Participants are students. The dataset contains the sensor data, survey responses, and educational data of the participants in a range of 2 to 3 months. What I'm focusing on in this project is the survey responses. These psychological surveys will assign scores according to the responses, revealing the extent of happiness, stress, and some aspects of personalities.

Each survey is corresponding to one measurement. The surveys includes PHQ-9 for depression, Perceived stress scale (PSS) for stress, Longliness scale, Flourish scale, Positive and Negative Affect Schedule (PANAS), and Big Five Personality Traits. After I aggregate the score of each survey for each student, I normalize the survey by rescale the score in the range of $[0, 1]$. For example, the Flourish scale have score range in $[8, 56]$. Given a student's score s, the normalized score is

$$\frac{s-8}{56-8} = \frac{s-8}{47}$$

After such preprocessing, the scores from different surveys are having the same scale. For the missing scores, a constant value -1 is filled in. The actual missing rate is approximately 20%. For the purpose of method evaluation, students without missing scores are extracted and will be used in the experiments, i.e. only the rest 80% known scores are used for train and validation.

5. Results

5.1. Preliminary Results

Table 1: Mean Square Error (MSE) of the imputation on validated survey responses, with 50% missing rate

Model	MSE
Mean Imputation	0.0122 ± 0.0013
Matrix Factorization	0.0122 ± 0.0029
MLP Autoencoder	0.0125 ± 0.0025
MLP-AE on Matrix Factorization	**0.0109 ± 0.0016**

In the preliminary experiment, I randomly generate 10 masks with 50% missing rate. As a result, the entries with value 1 are the training set, and the rest is the validation set. The final evaluation metric is the Mean Square Error on the validation set averaging across the 10 masks.

The baseline algorithm is the Mean Imputation, where the missing entries are replaced by the mean value of the known entries of the corresponding column. The size of latent space for both Matrix Factorization and MLP Autoencoder are set to 8. More specifically, Matrix Factorization will have $P \in \mathbb{R}^{N\times 8}, Q \in \mathbb{R}^{M\times 8}$, where N, M are the number of students and survey scores respectively. MLP Autoencoder will first encode the input $X \in \mathbb{R}^{N\times M}$ to $\mathbb{R}^{N\times 8}$, then decode it back to the input size. The algorithm that combine both Matrix Factorization and MLP Autoencoder have the same setting.

For all the algorithms except the baseline, Stochastic Gradient Descent (SGD) with 0.9 momentum is applied for parameter optimization. A constant step size 10^{-2} is used. The coefficient λ for the L2-norm regularization is set to 10^{-3}. These methods converge around 1000 epochs.

From the result table, we can draw the following preliminary conclusion: The method that combined both MLP Autoencoder and Matrix Factorization achieve the best performance (among these four methods); Matrix Factorization performs similar with the baseline, and imputation from plain Autoencoder results in relatively larger error.

5.2. Performance Under Different Missing Rate

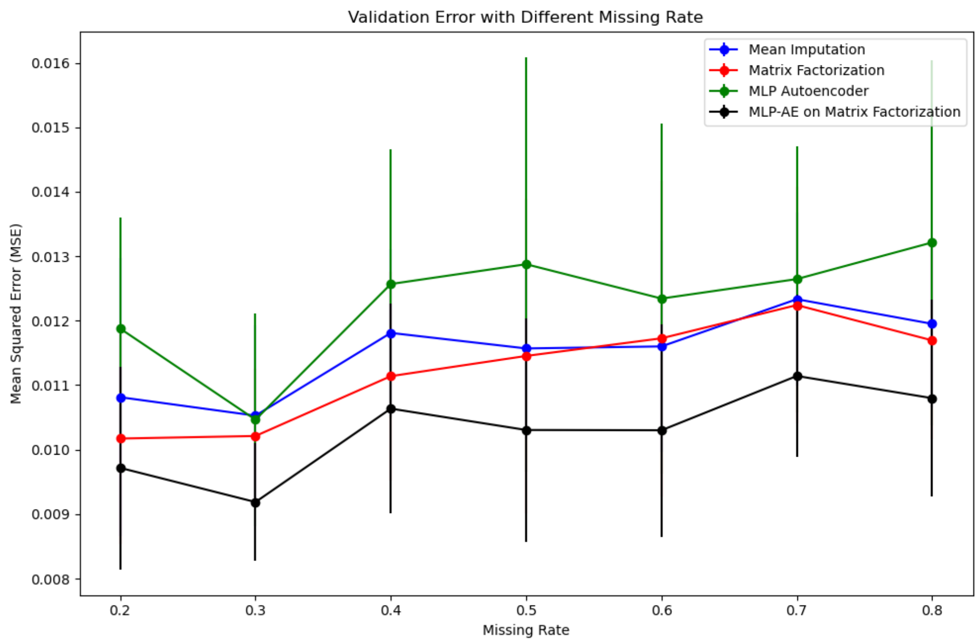

Figure 1: Validation Error under Different Missing Rate

This experiment tests the performance of the models trained on the dataset with different missing rates. More specifically, for each missing rate, the evaluation processes are same as the preliminary experiment. In the figure 1, we present the averaged Mean Squared Error (MSE) on the validation sets of each method with each missing rate, along with the error bar as the standard deviation.

There is a rough trend shows that the higher the missing rate, the errors are likely to be larger. Among these methods, the MLP Autoencoder with matrices from the method Matrix Factorization as embedding features always outperform other methods. The vanilla Autoencoder perform worse than the baseline. The vanilla Matrix Factorization method yields similar performance with the Mean Imputation method. Such results strengths the conclusions from the preliminary experiment.

6. Discussion

I've presented that MLP autoencoder that adopt Matrix Factorization as embedding achieve the best performance on the task of completion of survey scores in StudentLife dataset. The Matrix Factorization, as a collaborative filtering method, essentially benefit from making predictions according to nearest neighbors. Because this survey dataset is a small dataset, Mean Imputation is likely to have the same effect as applying collaborative filtering. The MLP autoencoder, on the other hand, is relatively hard to explain what is going on. As a deep learning method, we know that the autoencoder is trying to modeling the original data into a low-dimensional space, then reconstructing the true data from the latent representation. However, the original data contains many missing entries. A constant value is used to impute these entries before the data is fed for training. Such constant value act like a red-herring per se. This can be solved by adding a pre-imputation step, matrix factorization in my experiment, before the autoencoder. The factorized matrices are de facto the embeddings for both "student-survey" (or "user-item").

As the results shown in the previous section, the autoencoder does have ability to modeling the latent representation of the data, but it need preprocessing to remove its sensitivity to the initialized imputation.

7. Conclusion

In this project, I've experimented on the task of student survey scores imputation with collaborative filtering, deep learning, and a method that combine these two methods. The experiment results indicate that the vanilla MLP autoencoder didn't provide promising result. The collaborative filtering only outperform the baseline. The combined method that use collaborative filtering as embedding layer before the MLP autoencoder seems taking the benefit from both approaches, and achieve the best performance. As the current implementations of the collaborative filtering and the autoencoder are relatively light, I will try other structures for these sections in the future works.

References

T. Choi, H. K. Chan, and X. Yue. Recent development in big data analytics for business operations and risk management. *IEEE Transactions on Cybernetics*, 47(1):81–92, 2017. doi: 10.1109/TCYB.2015.2507599.

Jicong Fan and Tommy Chow. Deep learning based matrix completion. *Neurocomputing*, 266:540–549, 2017. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2017.05.074. URL https://www.sciencedirect.com/science/article/pii/S0925231217309621.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263.

P. Nagarnaik and A. Thomas. Survey on recommendation system methods. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pages 1603–1608, 2015. doi: 10.1109/ECS.2015.7124857.

Y. Park, S. Park, S. Lee, and W. Jung. Fast collaborative filtering with a k-nearest neighbor graph. In *2014 International Conference on Big Data and Smart Computing (BIGCOMP)*, pages 92–95, 2014. doi: 10.1109/BIGCOMP.2014.6741414.

Xiaoyuan Su and Taghi M Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009, 2009.

S. Taylor, N. Jaques, E. Nosakhare, A. Sano, and R. Picard. Personalized multitask learning for predicting tomorrow's mood, stress, and health. *IEEE Transactions on Affective Computing*, 11(2):200–213, 2020. doi: 10.1109/TAFFC.2017.2784832.

Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T. Campbell. Studentlife: Assessing mental health, academic performance and behavioral trends of college students using smartphones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, page 3–14, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329682. doi: 10.1145/2632048.2632054. URL https://doi.org/10.1145/2632048.2632054.