Hybrid Joke Recommendation System Report

Xuanjun Chen
Student ID: R09922165
Mail: R09922165@csie.ntu.edu.tw
Contribution: Algorithms development
Signature: Xuanjun Chen

Cheng-Wei Kao
Student ID: R09944030
Mail: R09944030@csie.ntu.edu.tw
Contribution: Algorithms development
Signature: Cheng-Wei Kao

Tsung-Lin Tsou
Student ID: B06902060
Mail: B06902060@csie.ntu.edu.tw
Contribution: Algorithms development
Signature: Tsung-Lin Tsou

Po-Yu Huang
Student ID: B06902034
Mail: B06902034@csie.ntu.edu.tw
Contribution: Demo web app development
Signature: Po-Yu Huang

1 INTRODUCTION

Reading jokes in life is a way to help us relieve stress. However, it is difficult to measure how funny a joke is. Everyone's cultural background is different, so some jokes are funny to some people, but not necessarily funny to others. Our motivation is to build a personalized joke recommendation system.

Then main contributions of this project are as follows:

1) We propose a Hybrid Joke Recommendation System (HJRS) by combining Latent Dirichlet Allocation method, Content-based filtering method and Matrix Factorization method.
2) HJRS has achieved state-of-the-art performance in all of our implemented models.
3) We have established a hybrid joke recommendation system demo website and both new and old users can get personalized recommendations.

The rest of this report is organized as follows. Section 2 shows the related work, and Section 3 provides the methods we implement, Section 4 presents the experiment results, and Section 5 concludes this project.

2 RELATED WORK

Compared to movie, book, or cloths recommender system, fewer research are dedicated to joke recommender system, but there are still some classical work study on joke recommendation. Goldberg *et al.* [1] proposed an Eigentaste algorithm, which uses principal component analysis (PCA) for dimensionality reduction and then clusters user in the lower dimensional subspace. The similarity between Eigentaste and HJRS is that our work also use dimensionality reduction in Matrix Factorization method and clustering in Latent Dirichlet Allocation method. But the difference is that our work also consider the content features and weighted ensemble multiple models.

3 METHODOLOGY

In this section, we will introduce all the single recommendation models that we have implemented. We also explain the hybrid mechanism used to come up with hybrid recommendations.

3.1 Single Recommendation Model

3.1.1 Popularity (POP) Recommendation. POP is the most naive baseline model. It uses the popularity score as the basis for recommendations. The popularity score of each joke is based on grouping all the jokes, and the average score of the rating obtained by each joke is counted as the popularity score of the joke. The baseline model will give priority to recommending the most popular joke to each user, which means that each user gets the same recommendation.

3.1.2 Graph Embedding (GE) Recommendation. Since we have the ratings between users and items, we can construct a bipartite weighted graph and get the user and item embeddings through random walk of graph. We refer to [3] to implement graph embedding methods, it proposed an edge-sampling algorithm to address weighted graph. The algorithm is to random walk to another that is connected to the current node. The probability depends on the ratings, which represents the weight of the edge. Then we will use skip-gram to find out the relations between users and items. The disadvantage is that the realations between user, user and item, item is not included. Otherwise, the graph can be more complete and the performance is expected to be enhanced.

3.1.3 User-based Collaborative Filtering (UBCF) Recommendation. The UBCF relies on the idea that users who have similar rating behaviours so far, share the same tastes and will likely exhibit similar rating behaviours going forward. In our task, the number of users is far more than the number of items. UBCF not only requires a lot of computing resources, but also cannot make any recommendations without the user's historical data.

3.1.4 Content-based Filtering (CBF) Recommendation. In this part, we obtain vector representation of each joke based on TF-IDF (Term Frequency-Inverse Document Frequency). Content based recommendation system doesn't involve other user's ratings but only his own ratings, so the predicted rating for a given user and item is obtained by weighted sum of the rating score of the top-k similar jokes given by that user. The pros of CBF recommendation is that only one rating is required for a new user to recommend him some similar jokes, and the cons is that the text features of the jokes sometimes cannot fully present the meaning of the jokes.

3.1.5 Latent Dirichlet Allocation (LDA) Recommendation. In this part, we obtain vector representation of each joke based on LDA. Since the joke metadata wasn't available, we first cluster 150 jokes into n clusters using this topic modeling technique, each cluster represents an abstract topic. Then we calculate the predicted rating for a given user and item based on weighted sum of the rating score of the top-k similar jokes in the same topic given by that user.

3.1.6 Matrix Factorization (MF) Recommendation. The MF model learns latent factors for each user and item and uses them to make rating predictions. Our implemented is based the MF technology summarized by Koren et al. [2]. Although the MF model can map a high-dimensional matrix to the product of two low-dimensional matrices, it solves the problem of data sparseness, but due to the large number of parameters that constitute the model, it is easy to overfit the data.

3.2 Hybrid Recommendation Model

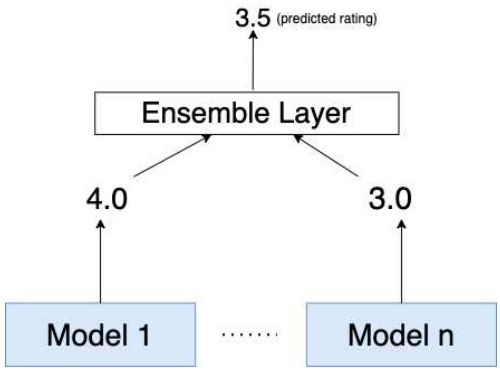

Figure 1: Example of hybrid recommendation model

The recommendation ability of a single recommendation model is always limited. In order to improve the performance of the model, we have integrated the output of multiple models, just like the example in Figure 1. In this project, we adopted two ensemble schemes, namely average ensemble (AE) and weighted ensemble (WE). Average ensemble is the result of directly averaging the prediction rating scores of all models. Weighted ensemble is to give each model a weight value, and the predicted rating score of every single model must be multiplied by the corresponding weight. Finally, we add up the rating scores of different models that have been multiply them by the weights to get the output of weighted ensemble. When using WE, if there are two models integrated, we will use equation (1) and if there are three models integrated, we will use equation (2).

$$\hat{y_i} = (1-\alpha) * model_{1_i} + \alpha * model_{2_i} \tag{1}$$

where $\hat{y_i}$ is the predicted value of the i-sample, α is the weight, $model_{j_i}$ is the predicted value given by model j.

$$\hat{y_i} = (1-2\cdot\alpha) * model_{1_i} + \alpha * model_{2_i} + \alpha * model_{3_i} \tag{2}$$

where $\hat{y_i}$ is the predicted value of the i-sample, α is the weight, $model_{j_i}$ is the predicted value given by model j.

4 EXPERIMENTS

4.1 Setup

The Jester Dataset 3 and Dataset 4[1] had collected 62,604 user ratings of 150 jokes. The rating scale is continuous -10.0 to +10.0, where -10.0 is the far end of the "Less Funny" direction, and +10.0 is the far end of the "More Funny" direction. However, some users rated jokes too few for us to recommend, so we only select users who have rated more than 10 jokes. We discard 15866 users and 46738 users remain. Then we select 90% of the jokes seen by each user as the training set and 10% as the test set. There are three columns for every single data, they are *user_id*, *joke_id*, *rating*. After prepossessing, we got 166,8432 data in the training set and 16,2954 data in the test set.

In order to make our results more reliable, we use 4 evaluation methods, they are Mean Squared Error (MSE), Normalized Mean Squared Error (NMSE), Mean Absolute Error (MAE), Normalized Mean Absolute Error (NMAE). Their mathematical formula is as follows:

$$MSE(y,\hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} (y_i - \hat{y_i})^2 \tag{3}$$

where $\hat{y_i}$ is the predicted value of the i-sample, and y_i is the corresponding true value, the MSE estimated over $n_{samples}$.

$$NMSE(y,\hat{y}) = \frac{MSE(y,\hat{y})}{\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \hat{y_i}^2} \tag{4}$$

where $\hat{y_i}$ is the predicted value of the i-sample, and y_i is the corresponding true value, the $NMSE$ normalize MSE by average of the power of predicted values.

$$MAE(y,\hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \hat{y_i}| \tag{5}$$

where $\hat{y_i}$ is the predicted value of the i-sample, and y_i is the corresponding true value, the MAE estimated over $n_{samples}$.

$$NMAE(y,\hat{y}) = \frac{MAE(y,\hat{y})}{\max_{y\in\hat{y}_i y} - \min_{y\in\hat{y}_i y}} \tag{6}$$

where $\hat{y_i}$ is the predicted value of the i-sample, and y_i is the corresponding true value, the $NMAE$ normalize MAE by the difference of maximum of predicted value and minimum of predicted value.

4.2 Performance comparison

Table 1 shows the performance of the models we have implemented. The upper part of the from are single recommendation models and the lower part are hybrid recommendation models. In the lower part, the plus sign (+) means to mix two models or three models. The AE is represent average ensemble and the WE is represent weighted ensemble. If the number of ensemble models is 2, then we use equation (1) to adjust the weighted. If the number of ensemble models is 3, then we use equation (2) to adjust the weighted.

As far as a single recommendation model is concerned, the MF model performed best, and its MSE, NMSE, MAE, and NMAE were 16.371, 0.956, 2.983, and 0.149, respectively. Both LDA model and CBF model are slightly weaker than MF model. The evaluations of

UBCF and GE at NMSE are 1.215 and 3.398 respectively, which are better than POP. However, UBCF and GE are very poor in each of MSE, MAE and NMAE, much worse than POP.

When it comes to hybrid models, the performance of all hybrid models in the three evaluation function of MSE, MAE and NMAE is better than the single LDA model and the single CBF model. The MF+LDA+CBF(WE) model performs best among all hybrid models. Its evaluation values for MSE, MAE, and NMAE are 15.968, 2.051, and 0.148 respectively. It has achieved state-of-the-art results in all models.

	MSE	NMSE	MAE	NMAE
POP	24.918	4.737	4.047	0.202
UBCF	32.408	1.215	4.434	0.222
GE	28.933	3.398	4.307	0.215
LDA	20.611	1.233	3.353	0.167
CBF	25.523	1.034	3.491	0.174
MF	16.371	**0.956**	2.983	0.149
MF + CBF (AE)	18.583	1.316	3.203	0.160
MF + LDA (AE)	17.531	1.519	3.214	0.161
MF + CBF + LDA (AE)	18.070	1.310	3.173	0.159
MF + CBF (WE)	16.155	0.972	2.954	0.148
MF + LDA (WE)	15.992	1.010	2.964	0.148
MF + LDA + CBF (WE)	**15.968**	0.991	**2.951**	**0.148**

Table 1: Performance comparison of different models.

4.3 Ablation Study

In this section, we will do ablation study for LDA model, MF model, and hybrid models, trying to find the key factors that make these model perform better.

4.3.1 LDA model. In the training process of LDA model, we need to assign how many topics the joke will be clustered into, hence we want to know which number of topics is the best choice for joke clustering. To achieve this goal, we perform the ablation study on the number of topics. According to the results in Table 2, we find that LDA model achieve the best performance when the number of topics equals to 2.

As the number of topics increases, it is not easy for the model to learn more general information. The relatively small number of topics can get better results.

	NO. of topics	MSE	NMSE	MAE	NMAE
LDA	2	**20.611**	**1.233**	**3.353**	**0.167**
LDA	3	21.196	1.257	3.394	0.169
LDA	4	22.101	1.286	3.463	0.173
LDA	6	23.203	1.344	3.552	0.177
LDA	8	24.476	1.373	3.634	0.181
LDA	10	24.652	1.392	3.662	0.183

Table 2: The ablation study result for LDA model on the number of topics.

4.3.2 MF model. Sometimes if we want to increase the number of latent factors, we often need more regularization to offset the variance introduced by more latent factors. Table 3 is the result of the ablation study about the number of latent factors in MF model. We find that MF model achieve the best performance in all evaluation matrices when the number of factors equals to 8. Table 4 is the result of the ablation study about the number of regularization parameter in MF model. We find that MF model achieve the best performance in most evaluation matrices when the number of regularization parameter equals to $1e-4$, except for NMSE.

This experiment shows that if the number of potential factors is too large, it may lead to over-fitting, and too few may lead to under-fitting. On the contrary, if the number of regularization is too small, it may lead to over-fitting, and if the number of regularization is too large, it may lead to under-fitting. We can only get better results by choosing a compromise value.

	NO. of Latent Factors	MSE	NMSE	MAE	NMAE
MF	2	17.732	1.963	3.318	0.166
MF	4	17.467	1.930	3.291	0.165
MF	8	**17.415**	**1.926**	**3.287**	**0.164**
MF	16	17.423	1.927	3.288	**0.164**
MF	32	17.421	1.927	3.288	**0.164**
MF	64	17.424	1.927	3.288	**0.164**

Table 3: The ablation study result for MF model on the number of factors.

	NO. of Regularization Parameter	MSE	NMSE	MAE	NMAE
MF	5e-2	24.832	5.058	4.044	0.202
MF	1e-3	17.425	1.928	3.288	0.164
MF	1e-4	**16.345**	0.956	**2.980**	**0.149**
MF	1e-5	17.719	0.928	3.092	0.155
MF	1e-6	17.849	**0.921**	3.099	0.155
MF	1e-7	17.97	0.927	3.112	0.156

Table 4: The ablation study result for MF model on the number of regularization parameter.

4.3.3 Hybrid model. Here we only discuss the hybrid model combining MF model, LDA model, and CBF model. The α in Table 5 comes from equation (2), which controls weight of the inside model in the hybrid model. Table 5 shows the results of ablation study when we take different values of α. We find that the hybrid model achieve the best performance in most evaluation matrices when the number of α equals to 0.1, except for NMSE.

This result shows that controlling LDA and CBF to add weighted ensemble in a relatively small ratio can improve the overall performance. If the ratio of LDA to CBF is relatively large, in the end, the entire prediction result will be dominated by these two models, and it will even lead to worse results than the original single model.

	α	MSE	NMSE	MAE	NMAE
MF+LDA+CBF(WE)	0.0	16.366	**0.954**	2.98	0.149
MF+LDA+CBF(WE)	0.1	**15.968**	0.991	**2.951**	**0.148**
MF+LDA+CBF(WE)	0.2	16.175	1.031	2.968	0.148
MF+LDA+CBF(WE)	0.3	16.987	1.070	3.026	0.151
MF+LDA+CBF(WE)	0.4	18.403	1.105	3.124	0.156
MF+LDA+CBF(WE)	0.5	20.425	1.132	3.261	0.163
MF+LDA+CBF(WE)	0.6	23.051	1.150	3.441	0.172
MF+LDA+CBF(WE)	0.7	26.282	1.161	3.658	0.183
MF+LDA+CBF(WE)	0.8	30.117	1.165	3.903	0.195
MF+LDA+CBF(WE)	0.9	34.558	1.165	4.172	0.209

Table 5: The results of the ablation study on the weighted number (α) of the hybrid model.

4.4 Demonstration

In addition to developing algorithms for jokes and verifying their performance, we also built a the Hybrid Joke Recommendation System especially for new users. A new user could rate the score of multiple jokes as shown in Figure 2. Figure 3 shows the top recommended joke for jester data. In our demonstration, we use hybrid (MF+LDA+CBF) as our metric.

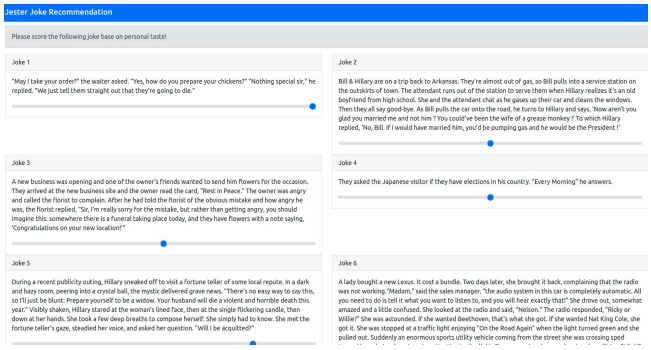

Figure 2: Joke hybrid recommendation system demo web application user interface.

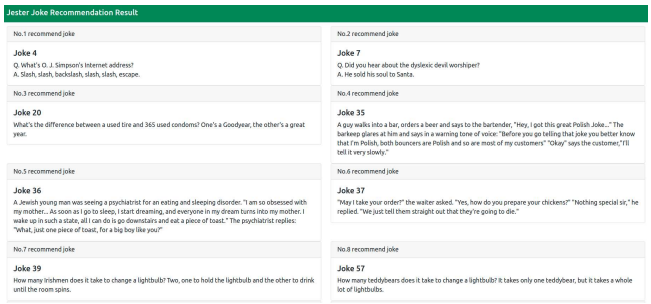

Figure 3: The result from our joke hybrid recommendation system.

5 CONCLUSIONS

We have three main contributions in this project: 1) We propose a HJRS model by combining LDA model, CBF model and MF model by weighted ensemble. 2) HJRS has achieved state-of-the-art performance in all of our implemented models. 3) We have established the HJRS demo website to recommend jokes to users. The most important thing is that we use a hybrid recommendation model, which greatly alleviates the problem of cold start, which means that both new and old users have the opportunity to obtain relatively personalized recommendations.

From this final project, we learn how to use user-based, content-based and graph embedding in real recommendation dataset. Among them, user-based outperforms and the others do not perform well since the number of jokes is not enough.

When it comes to future work, I think there are two aspects that can be improved. First, the weights between different sub-models in the hybrid model are weighted through adaptive learning. This method does not rely on the expertise of experts to manually adjust the weights. Second, we can do data augmentation by way of summary to solve the problem of extremely small number of jokes.

REFERENCES

[1] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. 2001. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval* 4, 2 (2001), 133–151.

[2] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[3] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.