Product Recommendation Based On Topic Modeling and Collaborative Filtering

Srinath Narayanan (A53098209)
and
Sanmitra Ijeri (A53105054)

1 Overview

We have built a product recommendation system using a joint probabilistic model based on topic modeling and collaborative filtering. To start with, we had implemented a so called 'naive' user based collaborative filter, in which we find the similarity between users, using a nearest neighbour approach based purely on the ratings they had given for different products. From these ratings, we deduce the **similarity scores** between users and use it to get a weighted average of ratings to fill in the ratings for products that the user hadn't reviewed yet (missing data). A recommendation would hence be made based on which of these newly rated items has a high rating, thereby recommending products preferred by similar users.

We further advanced this design by using topic modelling to leverage the plethora of information available in the review text of a review a user had given a product. We treat each of the reviews as individual documents and use LDA based topic modelling technique to find a distribution of documents over topics. We then perform sentiment analysis to see if the user has portrayed a positive or negative sentiment in the review and use it to convert the review's distribution over topics into scores for each review-topic pair. The scores for reviews from a single user are then averaged out to get user-topic scores. These scores give us more information about shared preferences between users and can be used to provide a more reliable nearest neighbour based **similarity scores** between users. Finally, these new similarity scores are used to generate a weighted average of ratings to fill in missing data, as seen before.

2 Literature Survey

One of the effective methods for collaborative filtering we came across is the low rank matrix approximation, which can be fitted to the data by MAP estimate of parameters. MAP estimate of parameters may, however, result in over-fitting of the model to the data. This can be handled by tuning of regularization parameters. Since tuning of regularization parameters is computationally intensive for large data sets, a Bayesian inference of the PMF model[3] would be the best way to perform collaborative filtering. This method also performs really well on extremely sparse matrices. The model capacity is controlled automatically by integrating over all model parameters and hyper-parameters, a variant of which uses adaptive priors over the 2 smaller matrices used to create the larger matrix.

But one valuable source of information that the BPMF model is not using is the reviews. Reviews give a great insight on what aspects of product a user actually cares about and how much they care about the same. By unraveling these aspects one can build a better recommendation engine- jointly modelling aspects, ratings and Sentiments as in [1], where topic modelling is used to extract such information while simultaneously calculating the rating and constructing the review for an unrated user-movie pair.

Since we have chosen to exploit the latent structure of the data using topic modelling (inspired from [1]), we have consequently chosen to use nearest neighbour based collaborative filtering to fill in missing data, instead of using BPMF (which models the latent features as well as fills in missing data). BPMF and naive collaborative filtering will be used to compare the performance of our model.

3 Collaborative Filtering with Topic modelling - our model

The 'Musical instruments' dataset we use for the experiment is such that none of the reviews are over 1500 characters. We are hence assuming that

- The author of the review conveys only a single sentiment in the review and doesn't convey a positive sentiment about a certain aspect (say, price) and a negative sentiment about another aspect (say, how long the product lasted/performance)

This modelling assumption (unlike in [1]) allows us to use a Sentiment analysis API[4] that returns an overall sentiment score between -1 and +1 for a single review. We then multiply this sentiment score with the probability distribution of the review over topics to get an idea of both.

- The degree of membership of a review with a topic
- whether it is because the user cares positively or negatively for the said topic

A brief perusal of the reviews in the dataset reveal this sort of structure to the reviews. We are hence leveraging a characteristic endemic to this domain to simplify a complex model like [1] to suit our needs.

The dataset[2] we use is also a '5-core' dataset - i.e, each user has reviewed at least 5 products. This makes it convenient to use a nearest neighbour based collaborative filtering approach along with topic modelling.

Overall, our model goes beyond the existing work by incorporating domain specific techniques alongside the best techniques already available in the existing literature.

3.1 Topic modelling using LDA

As mentioned earlier, the text in the user reviews, and not just the ratings, give a lot of insight about what aspects of a product a user likes or dislikes. By uncovering such aspects we get a better understanding of correlation between users and products. We will be using LDA based topic modeling to extract such features and combine this model with collaborative filtering to make more meaningful recommendations.

Given below is a DAG that explains our topic modelling assumptions.

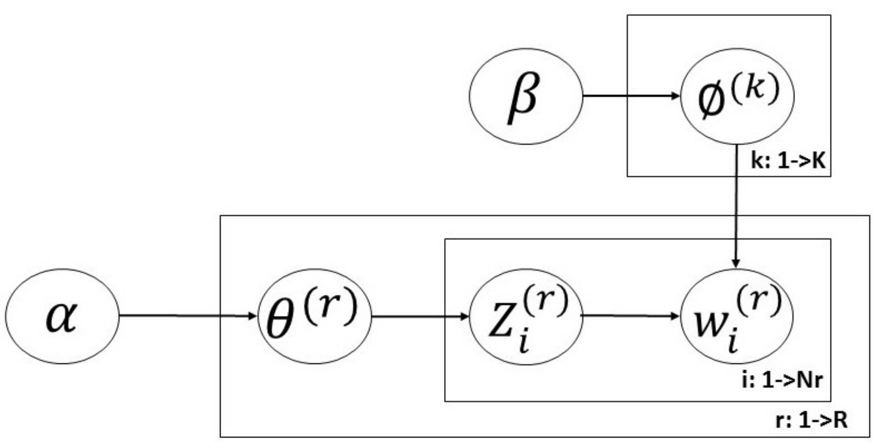

Figure 1: DAG for our LDA topic model

We treat each review as an independent document r, $1 \leq r \leq R$ where R is the total number of reviews present in the training set. $\phi^{(k)}$ is the probability distribution of topic k over the vocabulary of the training set and is a 1 x W vector, W being the size of the vocabulary. It has a Dirichlet prior parametrized by a uniform value of β, as is the norm to model any count vector. The probability distribution of the document over topics is represented by $\theta^{(r)}$, which is a 1 x K vector for K topics. $\theta^{(r)}$ also has a Dirichlet prior parametrized by a uniform value of α, to model the count vector. The i^{th} word of the r^{th} document is represented by $w_i^{(r)}$ and has a latent topic assignment of $z_i^{(r)}$.

The inference/learning method we use is the Gibbs sampling approach. The update rules for the different variables are given below:

$$
\begin{aligned}
p(z_i^{(r)} = k | z_{-ri}, w, \phi, \theta, \alpha, \beta) &= p(w_i^{(r)}, z_i^{(r)} = k | z_{-ri}, w_{-ri}, \phi, \theta, \alpha, \beta) \\
&= p(w_i^{(r)} | z_i^{(r)} = k, \phi^{(k)}) p(z_i^{(r)} = k | \theta^{(r)}) \\
&= \phi^{(k)}_{w_i^{(r)}} \theta_k^{(r)}
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
p(\theta^{(r)} | z, w, \phi, \theta_{-r}, \alpha, \beta) &= p(\theta^r | z^r, \alpha) \\
&= Dirichlet(\alpha + \textstyle\sum_{i=1}^{N_r} z_i^{(r)})
\end{aligned}
\tag{2}
$$

$$
\begin{aligned}
p(\phi^{(k)} | z, w, \phi_{-k}, \theta, \alpha, \beta) &= p(\phi^{(k)} | z, w, \beta) \\
&= Dirichlet(\beta + \textstyle\sum_{r=1}^{R} \sum_{i=1}^{N_r} z_{i,k}^{(r)} w_i^{(r)})
\end{aligned}
\tag{3}
$$

3.2 Sentiment Analysis and Collaborative Filtering

A flowchart of the modelling process after topic modelling (involving sentiment analysis and Collaborative filtering) is shown below.

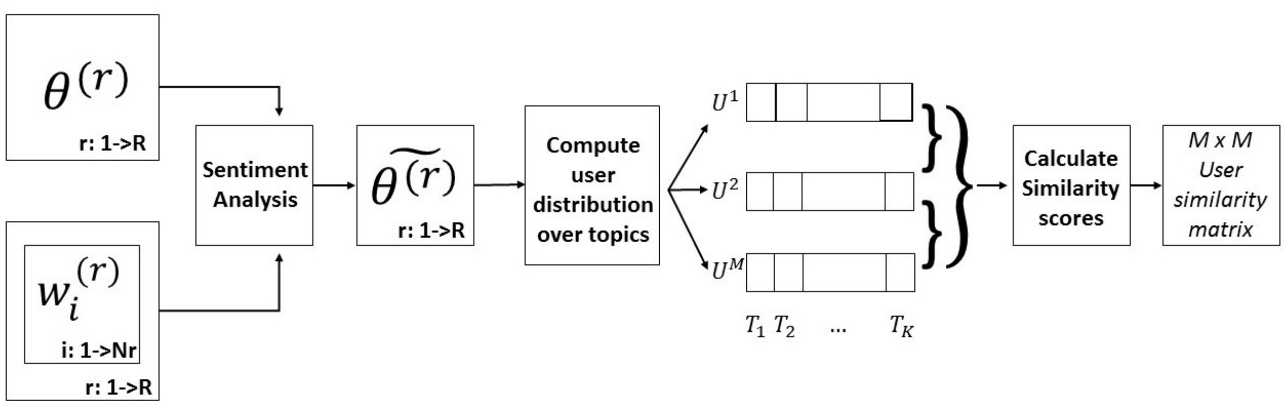

Figure 2: sentiment analysis and collaborative filtering flow chart

We use the Twinword API [4] to perform sentiment analysis on the text of each of the reviews individually. It returns a 'score' value for each review, with score $\in -1, +1$. This is then multiplied with each corresponding $\theta^{(r)}$ to create the 'scored distribution of reviews over topics' - $\widetilde{\theta^{(r)}}$

$$\widetilde{\theta^{(r)}} = \theta^{(r)} * \text{sentiment-score(r)}$$

Next, for each unique user, we find the user's distribution over topics U^i by averaging the $\widetilde{\theta^{(r)}}$ s where r is a review given by user i.

$$U^i = \sum_{r=1}^{R} \widetilde{\theta^{(r)}} \mathbb{1}(U(r) = U^i), 1 \leq i \leq M$$

where U(r) denotes the user who gave the review r and $\mathbb{1}(U(r) = U^i)$ is an indicator function to check that $U(r) = U^i$

We then calculate the similarity scores between users which is saved as a M x M User Similarity Scores (USS) matrix, such that each element of USS is given by

$$USS_{ij} = \frac{1}{\|\mathbf{U^i} - \mathbf{U^j}\|}$$

Since larger the euclidean distance between U^1 and U^2, the less similar U^1 and U^2 are. Hence there is an inverse relation between the user similarity scores and the euclidean distance between the User distributions over topics.

The Product rating matrix (PR) shows how N products have been rated by M different users. It is hence a N x M matrix of ratings, with initial entries for unrated (product,user) pairs being 0. The final matrices required for collaborative filtering are shown below.

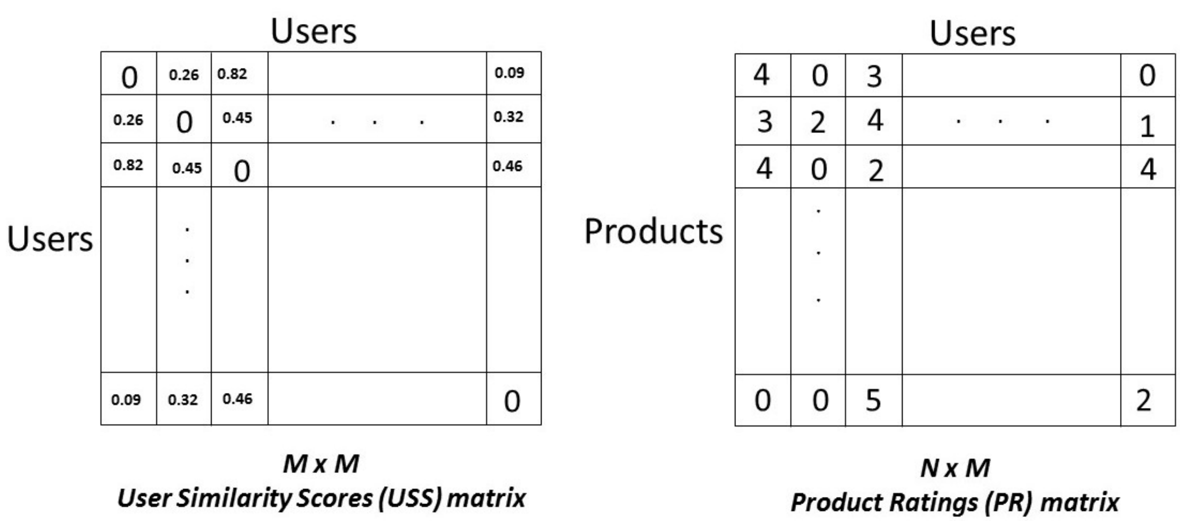

Figure 3: Final matrices used in collaborative filtering

A diagrammatic representation of collaborative filtering is shown below. The entries with 0 s are given the values of the weighted average of all other ratings for the product. The ratings that influence the predicted value of the element (which is initially 0) are represented by the blue arrows.

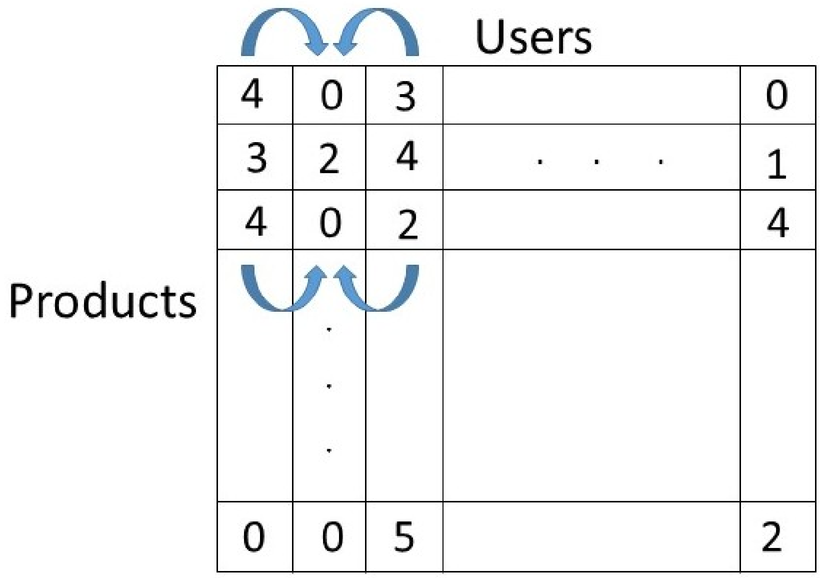

Figure 4: Diagrammatic representation of nearest neighbour based Collaborative filtering

We implemented a nearest neighbour based collaborative filter, in which we use the following intuitions

- The USS matrix gives us the similarity scores between users.
- Since the PR matrix is extremely sparse, we can't rely on just a 1-NN or 2-NN mechanism. We need to take all the other users or 'neighbours' into account.
- Hence for a user u, for each product, the new rating we provide will be the average weighted rating from all users

Hence, for a PR_{ij} element which is initially 0, the predicted rating is got by

$$PR_{ij} = \frac{PR_i \cdot USS_j}{\sum_{k=1}^{M} USS_{jk} \mathbb{1}(PR_{ik} > 0)}$$

4 Implementation

The dataset we used is the 5 core 'Musical Instrument reviews' dataset of 10,000 reviews (we will use the term **reviews** to refer to the **(user, product, rating, review text)** tuple) from the vast SNAP amazon reviews dataset[2], with each user having reviewed at-least 5 products. Among 10,000 reviews present in the data-set we had used 9,000 reviews for training and 1,000 for testing.

The Dirichlet parameters for topic modelling of our training data were set to $\alpha = 0.1$ and $\beta = 0.1$. To perform topic modelling, we first performed text vectorization on all the reviews in the dataset to create a vocabulary and built a document-term matrix for the same. Using this vocabulary and document-term matrix, we performed topic modelling with 2,4,5,10,15,20 and 30 topics to compare performance and finally choose the ideal number of topics. We then calculated the USS and PR matrix and filled in the missing entries in the PR matrix.

For comparison of performance, we tested the performance of BPMF, naive nearest neighbour collaborative filtering (a non-Bayesian technique which is for baseline comparison) and topic modelling + collaborative filtering (our model) on the above dataset and performed quantitative and qualitative assessment of the results as follows.

4.1 Results - Quantitative analysis of competing models and our model

To perform quantitative analysis, we calculate the Root Mean Square Error (RMSE) for all the 3 models for different settings of their parameters.

4.1.1 Naive nearest neighbour Collaborative Filtering

It was found that **RMSE = 1.11** between actual ratings and predicted ratings on the test dataset. For some users, we parsed and retrieved the review text for products rated highly by the user and for products both highly and poorly recommended by our model. A qualitative assessment of the reviews highlighted the usefulness of this model - Products rated highly by similar users were recommended (rating $\geq$ 4.5) and products rated poorly by similar users were not (rating $\leq$ 2.5).

This provides a useful baseline to compare the performance of our model against.

4.1.2 Bayesian PMF

We ran the Bayesian PMF model for 2,4,5,10,15,20,30 features and calculated the RMSE for each one of them. The model gave the lowest **RMSE = 0.89** for 5 features. It can be seen that arbitrarily increasing the number of features makes performance poor. This is intuitively understandable since the dataset is small and the actual number of features would be less (between 5 and 10).

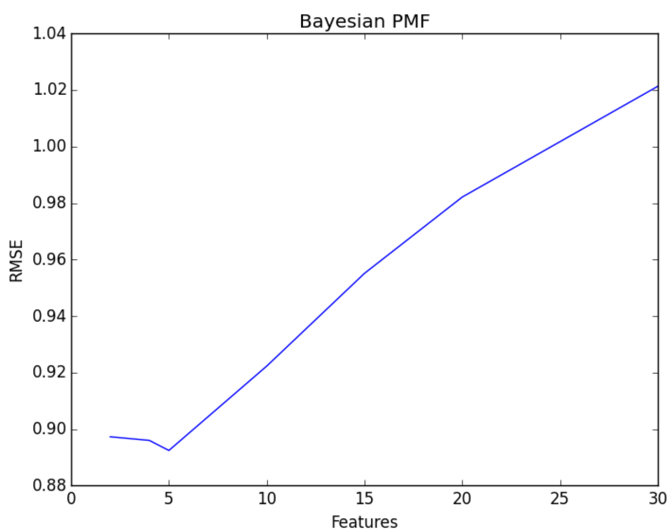

Figure 5: Plot of RMSE vs number of features for BPMF

4.1.3 Topic Modeling and Collaborative Filtering - Our model

We ran our model for 2,4,5,10,15,20 and 30 topics and calculated the RMSE for each one of them. The model gave very similar **RMSE $\approx$ 0.95** for each one of them, with the average value being. This intuitively makes sense, because similar users have similar topic distribution and euclidean distance between them based on their individual topic distribution does not vary much on the cardinality of topics.

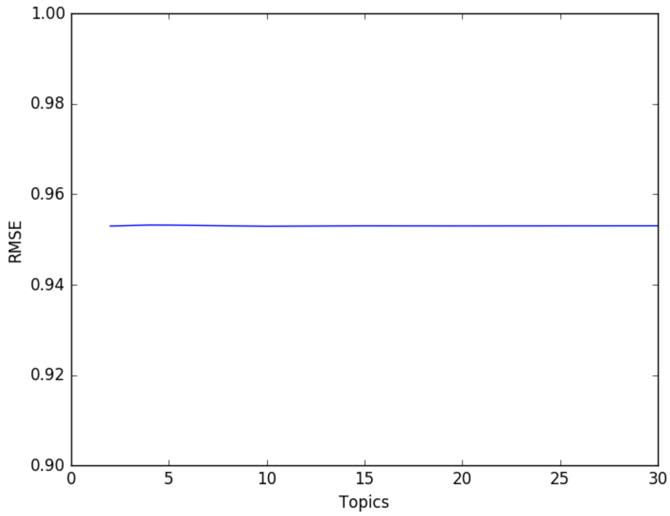

Figure 6: Plot of RMSE vs number of topics for our model

4.2 Results - Qualitative Analysis of our model

We chose K=10 for performing the following qualitative analysis.

4.2.1 Top 10 words in each topic

The following is a list of the top 10 words for each of the 10 topics. The topic names have been given after analysing the top words and, upon close inspection, seem internally consistent. For example, an aspect that most reviews deal with is the product price and/or cost efficiency which results in 'price' being one of the topics which almost all users are interested in. Other topic names are according to the actual type of product being purchased.

MICS	BATTERY / AMPS	CABLES	PRICE	GUITAR EXTRAS
Mic	power	cable	guitar	stand
sound	34	great	Just	guitar
recording	don	good	good	case
use	time	quality	Like	great
microphone	ll	cables	got	bag
good	battery	price	great	Just
studio	use	works	price	good
audio	way	use	bought	use
using	adapter	Pedals	really	Stands
mics	switch	pedal	product	does

GUITARS	GUITAR STRINGS	STRAPS / PICK	TUNERS	PEDALS
guitar	strings	strap	tuner	pedal
fender	sound	picks	guitar	amp
guitars	guitar	pick	use	sound
fit	string	guitar	capo	tone
neck	great	like	tune	like
strat	tone	just	easy	pedals
les	good	use	works	sounds
paul	like	good	string	great
screws	ve	great	tuning	just
bridge	acoustic	nice	like	really

Figure 7: Top 10 words in each topic after topic modelling

4.2.2 Topic distribution for different reviews

To analyse if the distribution of a review over topics makes sense, let us inspect the following sample review and its associated probability distribution. It has words that correspond to the 'mic' and 'price' topics and consequently the probability that this review belongs to those topics is higher.

Not much to write about here, but it does exactly what it's supposed to.
filters out the **pop sounds**. now my **recordings** are much more crisp.
it is one of the **lowest prices** pop filters on amazon so might as well **buy** it,
they honestly work the same despite their **pricing**

MICS	BATTERY/AMP	CABLES	PRICE	GUITAR EXTRAS	GUITARS	GUITAR STRING	STRAPS/ PICKS	TUNERS	PEDAL
0.355	0.205	0.005	0.355	0.55	0.005	0.005	0.005	0.005	0.005

Figure 8: Distribution of a review over topics

4.2.3 Sentiment analysis scores

To analyse if sentiment score makes sense, let us inspect the 2 following sample reviews and its associated sentiment scores - one review expresses a positive sentiment while the other expresses a negative sentiment.

- Review 1:
  - **TEXT:** "This pop filter is great. It looks and performs like a studio filter. If you're recording vocals this will eliminate the pops that gets recorded when you sing"
  - **SCORE:** 0.4075
- Review 1:
  - **TEXT:** "It hums, crackles, and I think I'm having problems with my equipment. As soon as I use any of my other cords then the problem is gone. Hosa makes some other products that have good value. But based on my experience I don't recommend this one."
  - **SCORE:** -0.1606

We can see that the first review with a positive sentiment is scored positively while the second review with the negative sentiment is scored negatively.

4.2.4 Similarity scores between users

To analyse the similarity scores between users, Let us take user 1 and his nearest neighbour user 2, who share a high similarity score of 0.89

Sample Reviews from User 1 for products with ids B001RNHE30 and B0002CZZW4 (not rated by user 2)

- Review 1:
  - **TEXT:** "I use the 21' cable. Great build quality, looks professional, sturdy plugs, and most importantly it sounds great! I like the smooth fabric sleeve, it doesn't stick to things like a plastic/rubber cable can"
  - **RATING:** 5.0
- Review 2:
  - **TEXT:** "These are high quality cables, both construction wise and in audio quality. I use these shorter cables if I have to run bits of my rack to external amps/units, ie: unplugging and plugging in a lot, and I find these cables stand up to more abuse than the average cable. I'm satisfied."

- **RATING:** 5.0

Sample Reviews from User 2 for products B000068NZC and B000VJJQUU (not rated by user 1)

- Review 1:
  - **TEXT:** "I buy this cable for my dj system and work great. i used a lot of time with my karaoke"
  - **RATING:** 5.0
- Review 2:
  - **TEXT:** "I like these short planet waves cables for short runs where there won't be a lot of abuse going on. They have good quality connector jackets similar to Monster cables. There are better made cables out there but few are in this price range."
  - **RATING:** 5.0

From the above reviews it is clear that, though both users have not rated the same products, both these users care about similar products (cables in this case). Hence their topic distribution is similar and similarity score is 0.89. i.e, 2 users with a high similarity score in our model are actually similar in terms of their interests in real life as well.

5 Conclusion

We have built a product recommendation system based on topic modelling and collaborative filtering, exploiting the information present in both user rating and reviews to give fine tuned recommendations. We used the non-Bayesian naive nearest neighbour collaborative filtering as a baseline method to test our performance on. We also compared the performance against the BPMF method present in existing literature to compare the performance of our model. Our model had a better RMSE score than the naive method, but was a little poorer than the existing, well studied BPMF technique. We also ran qualitative analyses to ensure that our topic modelling and collaborative filtering results made sense. We can conclude that performing topic modelling and using the latent structure present in the data can give significant boost to the performance of a recommendation system when compared to baseline techniques.

References

[1] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In SIGKDD. ACM, 2014.

[2] Julian McAuley, Rahul Pandey, Jure Leskovec (2015) Inferring Networks of Substitutable and Complementary Products. *in Knowledge Discovery and Data Mining*

[3] Ruslan Salakhutdinov and Andriy Mnih (2007) Probabilistic Matrix Factorization *in Neural Information Processing Systems Conference*

[4] https://market.mashape.com/twinword/sentiment-analysis-free - API link for sentiment analysis