Volume 3, Issue 9, September 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Hybrid Approach to Movie Recommender System

V.Adi Lakshmi, J.Kanaka Priya
M.Tech Scholar, Computer Science and Engineering
Gayatri Vidya Parishad College of Engineering(Autonomous)
Visakhapatnam ,Andhra Pradesh , INDIA

Abstract— Recommender systems provide users with personalized suggestions for products or services. Generally, recommender systems use Collaborative Filtering or Content-based methods to predict new items of interest for a user. Both methods have their own advantages, but individually they fail to provide accurate recommendations in many situations. A hybrid recommender system can overcome these shortcomings. In this paper, we present a hybrid approach for combining content and collaboration. We present experimental evidence using movie data to show that this approach improves recommendation accuracy.

Keywords— Collaborative Filtering, Content-based methods, Matrix Factorization.

I. INTRODUCTION

Recommender systems help to overcome information overload by providing personalized suggestions based on a history of a user's likes and dislikes. Popular recommender systems today are used by Netflix (www.netflix.com), Amazon (www.amazon.com), and Pandora (www.pandora.com) .They often take one of the following two approaches:

1. Collaborative filtering approach (Netflix): Based on the movies you liked and disliked, we have found users of similar tastes. Since they liked the following movies, we think you may like them too, even though we have no idea what types of movies they are [1].

2. Content-based approach (Pandora): Based on the songs you liked and disliked, it appears that you like slow, soft songs sung by a low, female voice. By analyzing all the songs in our digital library, these appear to be slow, soft songs sung by a low, female voice, and we think you will like them [1]

Content-based methods can uniquely characterize each user, but Collaborative Filtering has the advantage of (Herlocker *et al.* 1999)[4] not requiring machine analyzable content; thus it is capable of recommending an item without understanding the item itself (Su et al.)[6] But, for the very same reason, it suffers from "cold start" problem — predictions for newer items that have not received much user feedback tend to be very inaccurate. However, this problem can be mitigated to some extent by enhancing Collaborative Filtering to exploit any known content information. Melville et al. [4] developed such a hybrid, content-boosted Collaborative Filtering system by taking a two-step approach. They first filled in the sparse user rating matrix R with predictions from a purely content-based classifier, and then applied a Collaborative Filtering algorithm to the resulting dense matrix. In this paper, we propose a simple algorithm for incorporating content information directly into the matrix factorization approach (Koren et al[3]), which became popular due to the recent Netflix contest (www.netflixprize.com). Whereas Melville et al. [4] included content information as an intermediate step, we instead incorporated movie genre information directly as a natural linear constraint to the matrix factorization algorithm.

II. RELATED WORK

There have been a few other attempts to combine content information with collaborative filtering. One simple approach is to allow both content-based and collaborative filtering methods to produce separate recommendations, and then to directly combine their predictions (Cotter & Smyth 2000[3]; Claypool, Gokhale, & Miranda 1999)[2]. In another approach, Basu et al. (1998) treat recommending as a classification task.They use *Ripper*, a rule induction system, to learn a function that takes a user and movie and predicts whether the movie will be liked or disliked. They combine collaborative and content information, by creating features such as *comedies liked by user* and *users who liked movies of genre X*. Good et al. (1999)[8] use collaborative filtering along with a number of personalized information filtering agents. Predictions for a user are made by applying CF on the set of other users and the active user's personalized agents. Our method differs from this by also using CF on the personalized agents of the other users. In recent work, Lee (2001)[9] treats the recommending task as the learning of a user's preference function that exploits item content as well as the ratings of similar users. Our task is different from the previous works as we provide numerical ratings instead of just rankings. Our approach is more modular and general, and as such it is independent of the choice of collaborative and content-based components.

III. PROPOSED ALGORITHM

We will focus on the movie data from movie lens. We use movie genre as content information. Here is a summary list of some of our key notations:

- n_u – number of users;

- n_m – number of movies;
- n_g – number of genres;

•R – an $n_u \times n_m$ matrix, where each entry R_{um}, if not missing, is the score (in our case, an integer between 1and 5) given to movie m by user u;

•$\mathbb{L}$ – index of learning set, i.e., set of (u, m) such that R_{um} is observed (Fig. 1);

•$\mathbb{L}$m – set of u such that R_{um} is observed, for a given m;

•$\mathbb{L}$u – set of r such that R_{um} is observed, for a given u;

• $\mathbb{T}$ – index of test set, i.e., set of (u, m) such that R_{um} is observed but pretended to be "missing" by the collaborative filtering algorithms (Fig. 1);

• G– an $n_m \times n_g$ matrix, where each entry is defined as g_{mg} =1, if movie m contains genre g; 0, otherwise.

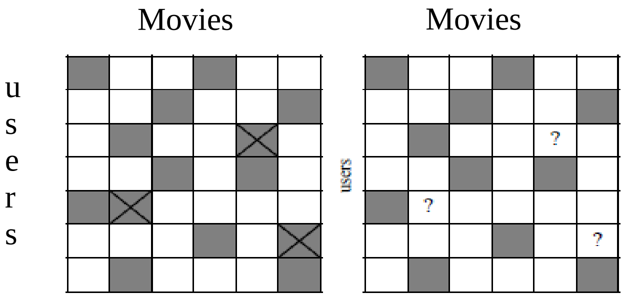


Figure 1: Experimental set-up.

Left: Some entries of R are observed (grey), while others are missing (white). Of the ob-served ones, 50% are randomly selected as test data (crosses) and treated as if "missing" by the algorithm. Right: The algorithm learns from all the observed (grey) entries and makes a prediction for every missing (white) entry, but we can only evaluate its performance (e.g., calculate the RMSE) using entries pretended to be "missing" (question mark).
Collaborative filtering algorithms work with the matrix R alone. The extra content information -genre for the movies is stored in the matrix G.

Experiences from the Netflix contest established the superiority of the matrix factorization approach [3] for collaborative filtering when dealing with large data sets. For a given dimension, n_f , the matrix factorization method aims to factor R into

$$R \approx U M^T = \underbrace{\begin{bmatrix}\mu_1^T\\ \mu_2^T\\ \vdots\end{bmatrix}}_{n_u \text{ x } n_f}\underbrace{[\rho_1 \quad \rho_2 \quad \cdots]}_{n_f \text{ x } n_m} \qquad (1)$$

where U is an $n_u \times n_f$ matrix whose u-th row is a feature vector $\mu_u \in R^{n}{}_{f}$ for user u, and M is an $n_m \times n_f$ matrix whose m-th row is a feature vector $\rho_m \in R^{n}{}_{f}$ for movie m. The objective is to find feature vectors, μ_u for each user u and ρ_m for each movie m, such that $R_{um} = \mu_u{}^T \rho_m$ estimates R_{um} .

This factorization can be achieved by considering the optimization problem,

$$\min_{U,M} \|R - UM^T\|^2$$

It is common to put regularization penalties on U and M in order to avoid overfitting, for example,

$$\min_{U,M} \|R - UM^T\|^2 + \beta (\|U\|^2 + \|M\|^2) \qquad (2)$$

However, since most entries of R are unknown, we can only compute the first Frobenius norm partially by summing over all known entries of R_{um}. This changes (2) into

$$\min_{\mu_u, \rho_m} \sum_{(u,m)\in\mathbb{L}} \left(R_{um} - \mu_u^T \rho_m\right) + \beta\left(\sum_{u=1}^{n_u} \|\mu_u\|^2 + \sum_{m=1}^{n_m} \|\rho_m\|^2\right) \qquad (3)$$

which is solved by an alternating gradient descent algorithm, moving along the gradient with respect to μ_u while fixing ρ_m and vice versa. That is, we iterate

$$\mu_u \leftarrow \mu_u + \alpha\left(\sum_{m\in\mathbb{L}_u} \left(R_{um} - \mu_u^T \rho_m\right) \rho_m - \beta \mu_u\right) \qquad (4)$$

$$\rho_m \leftarrow \rho_m + \alpha\left(\sum_{u\in\mathbb{L}_m} \left(R_{um} - \mu_u^T \rho_m\right) \mu_u - \beta \rho_m\right) \qquad (5)$$

until convergence, where α is the step size or learning rate, which we set sufficiently small to ensure convergence.

To exploit the extra content information genre G for the matrix factorization method, we constrained the feature vector of each movie to depend explicitly on its genre, i.e.,

$$M = G\,\emptyset \qquad (6)$$

where $\emptyset$ is an $n_g \times n_f$ matrix whose g-th row is a feature vector $\emptyset g \in R^n{}_f$ for genre g.

Under the constraint (6), model (1) became

$$R \approx U \emptyset^T G^T = \underbrace{\begin{bmatrix} \mu_1^T \\ \mu_2^T \\ \vdots \end{bmatrix}}_{n_u \text{ x } n_f} \underbrace{\emptyset^T}_{n_f \text{ x } n_g} \underbrace{[g_1 \quad g_2 \quad \cdots]}_{n_g x n_m}$$

which changed (3) into

$$\min_{\mu_u, \rho_m} \sum_{(u,m)\in\mathbb{L}} \left(R_{um} - \mu_u^T \emptyset^T g_m\right)^2 + \beta\left(\sum_{u=1}^{n_u} \|\mu_u\|^2 + \|\emptyset\|^2\right) \qquad (7)$$

Using the fact that $d(p^T Yq)/dY = pq^T$, we obtained the following alternating gradient-descent equations:

$$\mu_u \leftarrow \mu_u + \alpha\left(\sum_{m\in\mathbb{L}_u} \left(R_{um} - \mu_u^T \emptyset^T g_m\right) \emptyset^T g_m - \beta \mu_u\right) \qquad (8)$$

$$\emptyset \leftarrow \emptyset + \alpha\left(\sum_{(u,m)\in\mathbb{L}} \left(R_{um} - \mu_u^T \emptyset^T g_m\right) g_m \ \mu_u{}^T - \beta \emptyset\right) \qquad (9)$$

ALGORITHM

```
Step1:  $\emptyset$ = F.Transpose(), G=G.Transpose()
Step2:  $\hat{R}$ = product (U, $\emptyset$,G)
Step3:  Repeat Steps 4 to 9 for step= 1 to 5000
Step4:     Repeat Step5 for uidx= 1 to $n_u$

Step5:        Repeat Step6 for g= 1 to $n_g$
Step6:           Repeat Step7 for f= 1 to $n_f$
Step7:              Repeat Step8 for midx= 1 to $n_m$
Step8:                 if $R_{uidx,midx}$ > 0 then
                          e=  $R_{uidx,midx}$ - $\hat{R}_{uidx,midx}$
                          $U_{uidx,f} = U_{uidx,f} + \alpha * z(e*\emptyset_{f,g}*G_{g,midx} - \beta* U_{uidx,f})$
                          $\emptyset_{f,g} = \emptyset_{f,g} + \alpha * (e*G_{g,midx} * U_{uidx,f} - \beta*\emptyset_{f,g})$
Step9:        $\hat{R}$ = product(U, $\emptyset$,G)

Step10:    Return U, $\emptyset$.Transpose(), G.Transpose()
```

IV. Experimental results

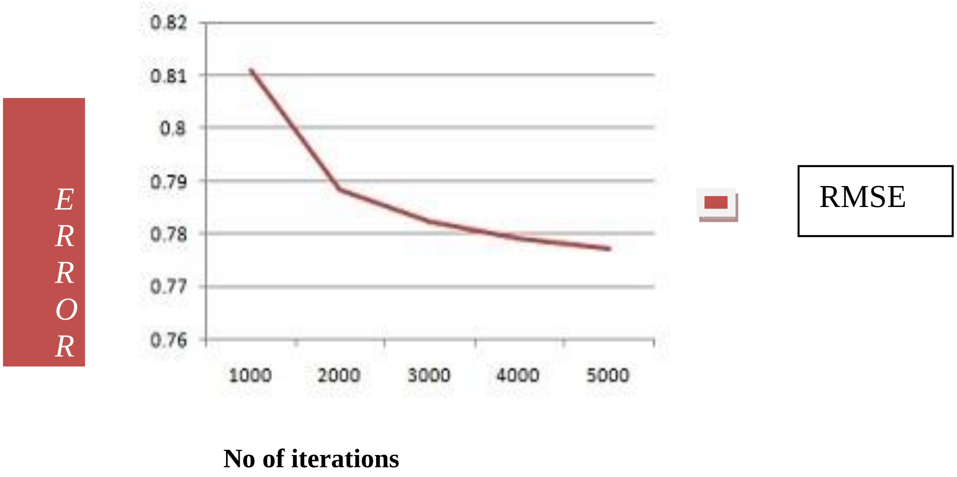
We conducted experiments on Movielens-100k dataset. It consists of 943 users, 1682 movies and 1 lakh ratings. We experimented with two features. For each experiment, 50% of the observed entries in R were randomly selected as test data, indexed by the set T. To evaluate the algorithms' performances, we followed Koren et al. [3] and used the root mean-squared error (RMSE) on T, i.e.,

$$\mathrm{RMSE}=\sqrt{\frac{1}{|\mathbb{T}|}\quad\sum\nolimits_{(u,m)\in\mathbb{T}}\left(\mathrm{R}_{um}\quad-\widehat{\mathrm{R}}_{um}\quad\right)^{2}}$$

where $\widehat{\mathrm{R}}_{um}$ denotes the score predicted by the algorithm under consideration, and |T| is the size of the test set T. Our experimental results are shown below.

No. of iterations	RMSE
1000	0.81092
2000	0.78844
3000	0.78222
4000	0.77912
5000	0.77728

Graphical representation is given in the following page.



After 5000 iterations there wasn't much improvement in RMSE.

V. Conclusion

The Netflix prize has rejuvenated a widespread interest in the matrix factorization approach for collaborative filtering. We described a simple algorithm for incorporating movie genre information directly into this approach. We conducted some experiments using movie data and confirmed that our Hybrid approach yields better results compared to Matrix Factorization technique. Collaborative filtering techniques do not give accurate results for new items because they are based on the ratings given by a set of users, which will be absent in this case. We overcome this problem by including content information. The ratings for the new movies are predicted based on their genre information. The accuracy of Hybrid technique can be further enhanced by considering the demographic information of the users along with the genre information of the users.

REFERENCES

[1]. Peter Forbes, Mu Zhu Content-boosted Matrix Factorization for Recommender systems: Experiments with Recipe Recommendation. *RecSys'11,* October 23–27, 2011.

[2]. Claypool, M.; Gokhale, A.; and Miranda, T. 1999. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of the SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation*.

[3]. Cotter, P., and Smyth, B. 2000. PTV: Intelligent personalized TV guides. In *Twelfth Conference on Innovative Applications of Artificial Intelligence*, 957–964.

[4]. Herlocker, J.; Konstan, J.; Borchers, A.; and Riedl, J. 1999. An algorithmic framework for performing collaborative filtering. In *SIGIR '99: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 230–237.

[5]. Melville, P., Mooney, R. J., and Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendation. In Proceedings of the 18th National Conference on Artificial Intelligence, pages 187–192.

[6]. Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. Advances in Artificial Intelligence, 2009. Article ID 421425.Basu, C.; Hirsh, H.; and Cohen, W. 1998.

[7]. Recommendation as classification: Using social and content-based information in recommendation. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*,714–720.

[8]. Good, N.; Schafer, J. B.; Konstan, J. A.; Borchers, A.; Sarwar, B.; Herlocker, J.; and Riedl, J. 1999. Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, 439–446.

[9]. Lee, W. S. 2001. Collaborative learning for recommender systems. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML-2001)*, 314–321. Mitchell, T. 1997. *Machine Learning*. New York, NY: McGraw-Hill.

[10]. Popescul, A.; Ungar, L.; Pennock, D. M.; and Lawrence, S. 2001. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the Seventeenth Conference on Uncertainity in Artificial Intelligence*.