

How to Get Endorsements? Predicting Facebook Likes Using Post Content and User Engagement

Wei-Fan Chen, Yi-Pei Chen and Lun-Wei Ku

Institute of Information Science, Academia Sinica, Taipei, Taiwan

{vieficwf, ypc82, lwku}@iis.sinica.edu.tw

Abstract. We view the prediction of Facebook likes as a content suggestion problem and show that likes can be much better predicted considered post content or user engagement. Experiments presented are based on a dataset of over 4 million likes collected from over seventy thousands of users in fan pages. The proposed model adopts the similarity metric to appraise how a user may like a document given his or her liked documents, as well as the Restricted Boltzmann Machine (RBM) to estimate whether a user would like a document given the like records of all users. The model achieves a precision of 5-10% and a recall of 2-55%. The commonly used label propagation model is implemented and tested as a baseline. Different models and settings are compared and results show superiority of the proposed models.

Keywords: Recommendation System, RBM, BLEU, Facebook.

1 Introduction

It has been shown that Facebook likes, as a kind of easily accessible digital records of behavior, can be used to automatically and accurately predict a range of highly sensitive factoid and opinionated personal attributes, including age, gender, parental separation and political views, stance, happiness, respectively [1]. However, literature shows that most researches use real likes instead of predicting them [2], which makes the information of a large quantity of new documents difficult to be utilized.

Predicting Facebook likes is to predict "who will read and like the current document enough to push the like button". It is similar to the product recommendation problem, which predicts "who will view and like the product enough to purchase it". However, there are still several differences between them, and the major difference is the degree of willingness. For the product recommendation research, there are usually view records and purchase records showing two different degrees of willing for utilization. View records indicates the relevance and slight preference, while purchase records show the strong preference (to the degree of willing to spend their money). Instead in Facebook, we have only like records, whose degree of willing is between view and purchase records. It costs some effort to push the like button, but the effort is little compared to the purchase. The difference in degrees of willing leads to different challenges.

In Facebook, social network should help a lot in predicting likes. However, the privacy issue has become crucial for all social media and the social network, e.g., friend information is not easily accessible anymore. Many researches ask volunteers to access their social network, but this is not feasible in real applications.

In this paper, we solve the Facebook like prediction problem with only the public information: the document content and the user engagement to the document. Therefore, we divide our research problem into two sub-problems: predicting likes by document content and predicting likes by user engagement. The former is realized by calculating similarities between documents with the intuition that users should like to read similar documents, i.e., documents of preferred topics or viewpoints. The latter is realized by a Restricted Boltzmann Machine (RBM) as it is shown to have good prediction power with enough known records [3]. Then we conquer this problem with a weighted function considering these two sub-models. Through the prediction of likes, we hope to provide valuable reference for the government and the companies to get more endorsements by their documents.

2 Related Work

The recommendation problem has been widely studied for decades [3]. It is known as collaborative filtering (CF) and researchers divide them into item-based, user-based and hybrid according to their recommendation strategy.

For item-based CF, it recommends items that similar to user's previous liked items or purchased products [4, 5], some of them has been adopted on business like Amazon [4]. The key point of item-based CF is presenting each item as a vector and then performing an item-to-item similarity matrix. For example, the item vector in Linden's work is an M dimensional vector, where each dimension correspond to a customer [4]. Also, there are ways to compute the similarity between items, e.g., cosine similarity, *Pearson-r* correlation, or the adjust cosine similarity where the values in each dimension of the vector are normalized firstly [5]. Besides product recommendation, document recommendation is similar to our work. For example, PRES (Personalized REcommender System) used TF-IDF of each word as a document vector and utilized cosine similarity to recommend web posts based on content [6]. The major difference between PRES and our work is that we focus on predicting the "like" behavior on Facebook but PRES aimed at retrieving relevant (judged by user feedback) hyperlinks to users.

For user-based CF, items that liked by similar users will be recommended. It can be performed using the same ideology in item-based CF: presenting a user as a vector according to her liked/purchased record and then computing a user-to-user similarity matrix. However, the shortage of using user vectors instead of item vectors is the sparsity of the user vectors because the size of users is significant larger than the size of items. To deal with this problem, researchers added more information on how to compute the similarity between users, e.g., social relationship [7], similar interests [8] or similar video viewing records [9]. Researches using user-based CF have shown that user information benefits the recommendation for product [7], document [8, 10, 11], and even video [9]. Besides, some researches in social network domain use belief propagation for marketing [12, 13], which can be considered as a variation of user-based CF to utilize the user information.

As for hybrid model, it combines item-based and user-based information to recommend [14, 15]. However, the problem turns to the combination of two information as Melville addressed [14]. In this paper, we propose a late-fusion approach, which combines two information via a weighted function.

3 Method

As mentioned, we propose a hybrid model that incorporates probabilities from the item-based model and the user-based model. Given a set of users and a set of documents, the likes form a matrix $\mathbf{L} \in \mathcal{R}^{U \times D}$, where one column indicates one user and one column indicates one document. The matrix \mathbf{L} is a sparse matrix where the elements $l_{i,j}$ in \mathbf{L} equals to one if user u_i liked the document d_j in the dataset. In fact, the like prediction is to estimate the “unknown” elements in \mathbf{L} , where the user u_i did not click like to document d_j . In our paper we aim at estimating the $l_{i,j}$ using item-based method and user-based method. Assuming that they can be independently estimated, formally we write,

$$p(l_{i,j}) = w \cdot p_{user}(l_{i,j}) + (1-w) \cdot p_{item}(l_{i,j}) \quad (1)$$

where $p(l_{i,j})$ is the estimated value of $l_{i,j}$, $p_{user}(l_{i,j})$ and $p_{item}(l_{i,j})$ denote the probability estimated by the user -based and item -based model, respectively, and w is used to control the weights of the two models.

In the following, we will detail how we compute two probabilities based on different features.

3.1 Item-based Model: BLEU

As for item-based model, we recommend documents that similar to the documents that the user liked. There are two assumptions used in our item-based model. First, users have no preference among documents, that is, users equally liked them. Second, we have very long term liking history that logs likings of the user and thus all kinds of liked documents were known, then we can model the liking probability via similarities. With this in mind, given a document d_j , we compare d_j with what user liked before, and choose the highest similarities among all comparisons as the estimated probability. Formally,

$$p(l_{i,j}) \cong \max \left(sim(d_j, d_m) \right), \forall d_m \in D_i \quad (2)$$

where d_j is the target document, D_i is the documents that user u_i liked, d_m is one of document in D_i , and $sim(\cdot)$ is the similarity function. In this paper, we approach the similarity function by the BLEU score.

BLEU is a modified form of precision which compares the candidate translation against multiple reference translations. It is a commonly used measurement for the quality of bilingual machine translation considering the similarity of n -grams between the candidate translation and the reference translations [16]. To predict whether a user u_i will like the current document d_j , we view it as the candidate translation of the reference translation set D_i , which includes all kinds of documents this user has ever liked in the whole dataset. That is, the document d_j is an alternative way to express the same content of one document in D_i . As a result, if the translation quality is high enough to indicate that d_j can be derived from D_i , user u_i would like document d_j with the estimated probability. To calculate BLEU scores, documents are tokenized into words and all punctuations are removed. However, unlike machine translation, the topics of the reference documents may vary. Therefore, we calculate BLEU scores between d_j and each reference document d_m , and report the BLEU score by maximizing these BLEU scores.

Then for each user u_i , we sort the probabilities among the documents not in D_i and recommend the top n documents to the user.

3.2 User-based Model: RBM

For user-based model, we recommend documents liked by other users that similar to the target user. The main challenge in presenting a user as a vector is the sparsity of the like pattern of a user, where a like pattern is a high dimensional vector and there are only a few of ones but most of the values are zero. In our paper, we use a simple deep learning model –Restricted Boltzmann Machine (RBM) to encode a like pattern into a low-dimensional and dense vector as shown in Figure 1 [11].

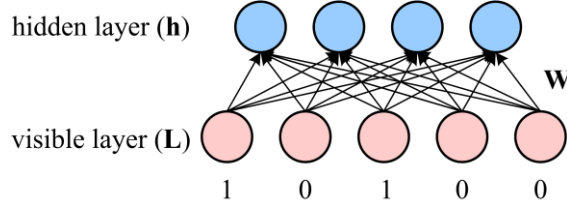


Fig. 1. A sample RBM network

Formally, for each user, the like pattern is a binary vector $\mathbf{L}_i = \{l_{i,1}, l_{i,2}, \dots, l_{i,j}, \dots, l_{i,|\mathbf{D}|}\}$, where $l_{i,j} = 1$ if the user like the document d_j , and $|\mathbf{D}|$ is the number of documents in our dataset. Then the RBM aims to optimize the parameters that maximize the observed probability $p(\mathbf{L}_i)$, shown in Equation 3.

$$\theta = \arg \max_{\theta = \{\mathbf{W}, \mathbf{b}_L, \mathbf{b}_h\}} \ln p(\mathbf{L}_i) \quad (3)$$

where \mathbf{W} , \mathbf{b}_L and \mathbf{b}_h are weights, bias for visible layer and bias for hidden layer, respectively. Then the RBM model optimizes the joint probability of the visible layer \mathbf{L}_i and the hidden layer \mathbf{h} in Equation 4, where the energy function is defined in Equation 5.

$$p(\mathbf{L}_i, \mathbf{h}) = \frac{1}{\mathbf{Z}(\theta)} e^{-\mathbf{E}(\mathbf{L}_i, \mathbf{h})} \quad (4)$$

$$\mathbf{E}(\mathbf{L}_i, \mathbf{h}) = - \sum_{l_{i,j} \in \mathbf{L}_i, h_k \in \mathbf{h}} l_{i,j} h_k W_{j,k} - \sum_{l_{i,j} \in \mathbf{L}_i} l_{i,j} b_{l_j} - \sum_{h_k \in \mathbf{h}} h_k b_{h_k} \quad (5)$$

After the training process, we used the parameters θ to predict likes. The one step Gibbs sampling was adopt to approximate the probability of a user to like the document as in Equation 6 and 7, suggested in [11].

$$\theta = \arg \max_{\theta = \{\mathbf{W}, \mathbf{b}_L, \mathbf{b}_h\}} \ln p(\mathbf{L}_i) \quad (6)$$

$$p(l_{i,j} = 1 | \hat{p}_i) = \sigma(b_{l_j} + \sum_{\hat{p}_k \in \hat{p}_i} \hat{p}_k W_{j,k}) \quad (7)$$

where $\hat{p}_k \in \hat{p}_i$ is the probability of the hidden layer given the visible layer, and $p(l_{i,j} = 1 | \hat{p}_i)$ is the like probability for the document d_j of the user given the hidden layer. We then use the probability in Equation 7 to recommend the top n documents for the user.

4 Experiments

4.1 Dataset

The experimental dataset was collected from Facebook fan pages related a same topic — nuclear power. The posting time of documents spans from September 2013 to August 2014. A total of 34,402 documents as well as their author and liker IDs were recorded. For content-duplicated documents (usually re-posts), their authors and likers were merged. Although the posting and liking behavior might have different implications, we default that authors should like whatever they posted. In addition, we removed users and documents having fewer than ten likes, and randomly selected 10% of likes per user as the testing data and 90% of likes per user as the training data. Table 1 shows the result after removing these users and documents.

Table 1. Like statistic

#documents	#users	#likes/doc	#likes/user	#likes (training)	#likes (testing)
34,402	77,416	123.48	54.87	3,821,359	426,609

4.2 Baselines

We use label propagation as a baseline to find potential likes. Given the like matrix \mathbf{L} , a transition matrix \mathbf{T} that defines the label transition probability from document to document, the goal of label propagation is to update \mathbf{L} given \mathbf{T} . Formally,

$$\mathbf{L}' = \alpha \cdot \mathbf{L}^0 + (1 - \alpha) \cdot \mathbf{T} \cdot \mathbf{L} \quad (8)$$

where \mathbf{L}^0 , \mathbf{L} and \mathbf{L}' are the prior label, the label from previous iteration and the updated label; \mathbf{T} is the transition matrix, and α is the prior parameter that determines the initial label priority. Note that the label here denotes whether the user likes the document. We repeatedly compute Equation (8) to update \mathbf{L} and predict new likes. Different factors are considered in building the transition matrix \mathbf{T} : co-liker, semantic similarity using BLEU score or n -gram vector. That is to say, a higher probability to be liked by a certain user are assigned to two documents that are liked by similar users or use similar words. Co-like \mathbf{T}_L is calculated by the *Jaccard* coefficient where $U(d)$ is liker set of document d :

$$\mathbf{T}_L(i, j) = \frac{|U(d_i) \cap U(d_j)|}{|U(d_i) \cup U(d_j)|} \quad (9)$$

For semantic similarity using BLEU score \mathbf{T}_B , we use the BLEU score calculated in the section 3.1. On the other hand, for semantic similarity using n -gram \mathbf{T}_g , we first present a document with a binary vector $v(d) \in R^{|\mathbf{V}|}$ using n -gram features, where $|\mathbf{V}|$ is the size of n -grams in our dataset, including uni-gram, bi-gram and tri-gram. Then for any two documents, we compute the cosine similarity using their n -gram vectors to form the transition matrix.

4.3 Evaluation Metric

For each user u_i , assuming a set of documents D_i was liked by u_i in the testing data, we recommend top n documents D_n for the user (likes in the training data will not be recommended). The precision is given by $|D_i \cap D_n|/|D_n|$, and the recall is given by $|D_i \cap D_n|/|D_i|$. In the results section, we report the user average precision and the user average recall on top n documents. We plot the ROC curves (Receiver Operating Characteristic) and the PRT curves (Precision-Recall-Threshold) for further discussion.

5 Predicting Facebook Likes

We first demonstrate the performance of prediction of the item-based model and the user-based model. Then, the result of the hybrid probability model is reported.

5.1 Item-based Model

For item-based model, documents that have similar content would be recommended to be liked by the same set of users. The PRT curve in Figure 2 shows the precision and recall when recommending different numbers of documents per user. The limitation of item-based model on Facebook data can be found from Figure 2. Recalls are lower than 5% and precisions are lower than 0.5%. Owing to the fact that the documents in our dataset are all related to a same topic, the recommendation system based on semantic features is hard to predict like. Even though some documents are semantically similar to what a user liked before, the documents that hold opposite opinion might not interest the user.

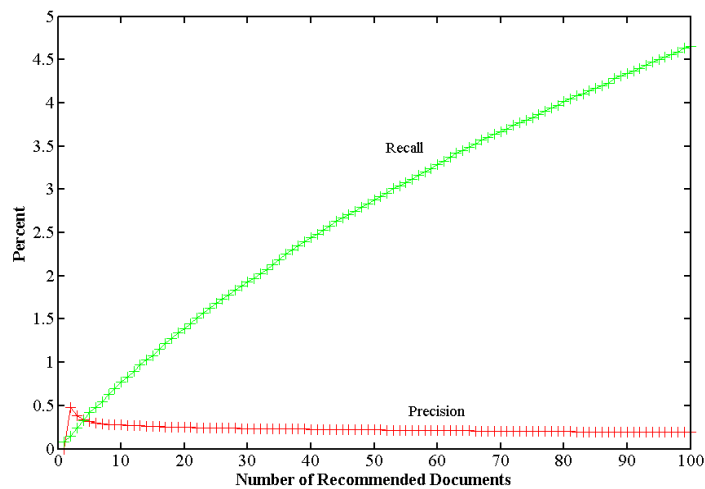


Fig. 2. PRT for item-based model

5.2 User-based Model

In the user-based model, we generate the probability for each document and user pair. The same set of documents would be recommended to users having similar like- records. The PRT curve in Figure 3 shows that user-based model has higher recall but lower precision. This tendency can also be found in the related work, where the user-

based model has precision in the range of 2-10% while recall in the range of 3-16%. However, our user-based model using RBM achieves higher precision (in the range of 5-10%) and recall (in the range of 2-55%). Comparing with item-based model, the performance of user-based model is significantly better. It suggests that user-based model successfully captures the taste of users and predicts the interesting documents to the target audiences.

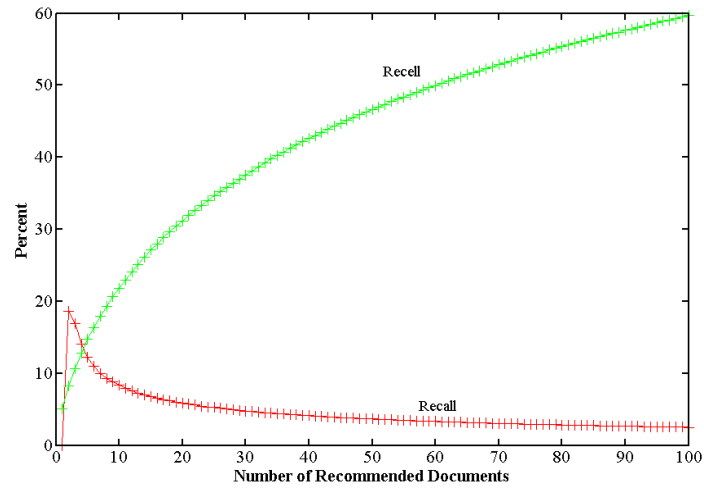


Fig. 3. PRT for user-based model

5.3 Hybrid Model

For the hybrid model, we joint the item-based model and the user-based model, setting w as 0.5 (which means item-based and user-based model are equally important). The PRT curve is shown in Figure 4. We find that the worse results of item-based model propagate noise to the hybrid model. It suggests that our late-fusion method using weighted function would be seriously harmed by noisy components.

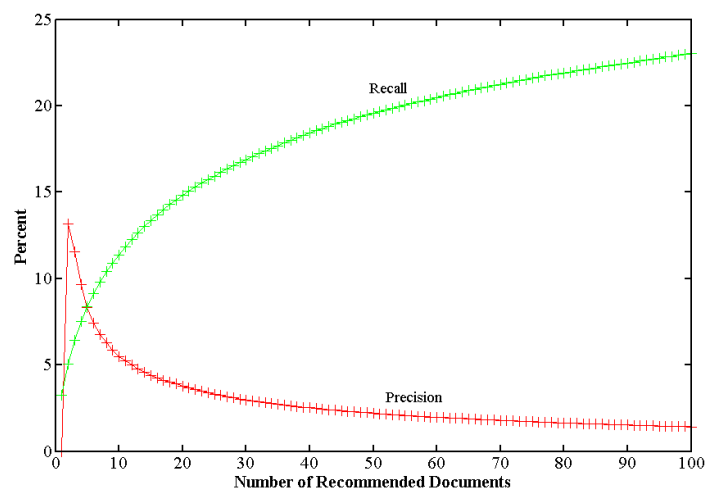


Fig. 4. PRT for hybrid model

5.4 Discussion

For comparison, we implement the label propagation, which has been widely used in classification [17] or recommendation problems [9]. Figure 5 shows the PRT curves of label propagation with co-liker feature and Figure 6 shows the ROC curves of label propagation with co-liker feature and our user-based model using RBM. Comparing with RBM model, the label propagation is based on the same information but the results in Figure 6 show that the proposed model successfully utilizes the visible units of RBM, and largely improves the performance. The label propagation does not perform well on the like prediction problem especially because of the sparsity of the like matrix. However, the user-based model using RBM can encode the user information into a dense vector and decode it to generate the like prediction.

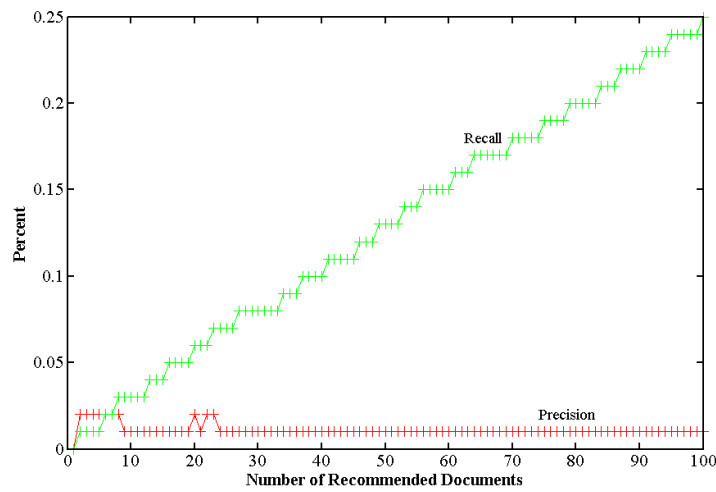


Fig. 5. PRT for label propagation (co-liker)

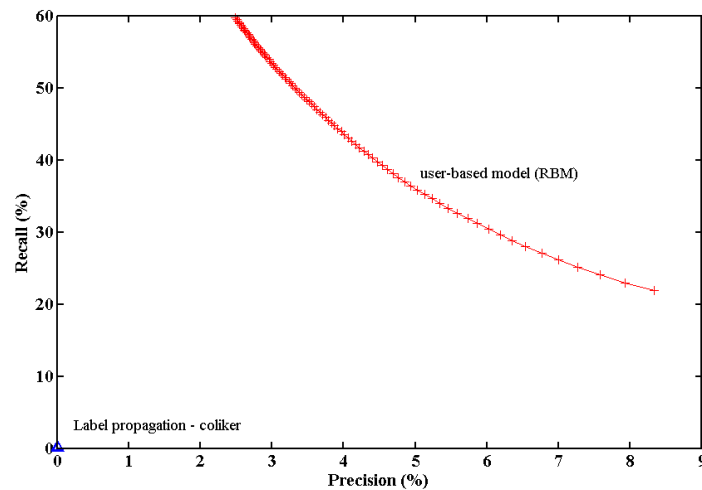


Fig. 6. ROC for RBM and label propagation with co-liker

Figures 7 and 8 show the PRT curves of label propagation with the n -gram vector similarity and the BLEU score, respectively. Figure 9 shows the ROC curves of above label propagation models and our item-based model.

From Figure 7 and 8, we can tell that the like prediction methods using semantic features are limited. Besides, the BLEU score is slightly better than simple cosine similarity of n -gram vectors.

Figure 9 shows that the strategy to find documents which are similar to the liked documents is better than propagate like information among a semantic similarity matrix. Receiving the like information from the most similar documents would be better than receiving the information from all documents. This phenomenon can be found in some related work where they suggested finding and propagating information to the nearest neighbors (four nearest neighbors in their paper) [18].

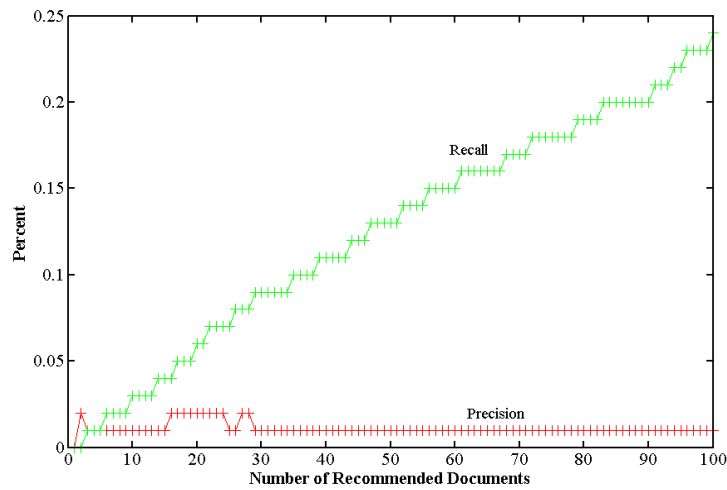


Fig. 7. PRT for label propagation (n -gram vector)

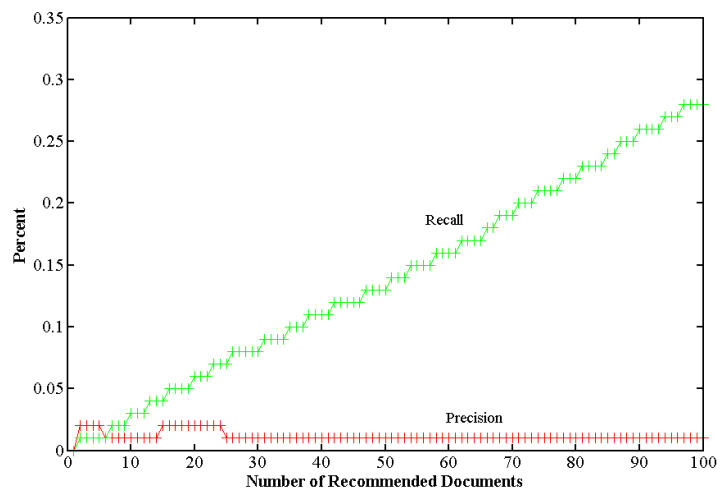


Fig. 8. PRT for label propagation (BLEU score)

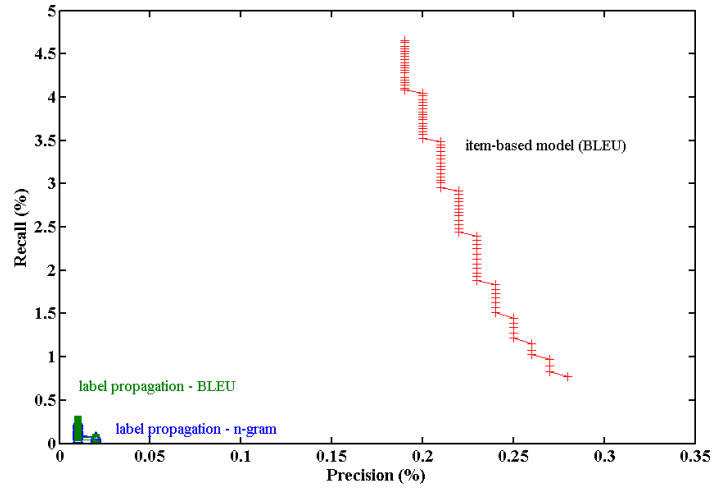


Fig. 9. ROC for item-based model, label propagation with n -gram vector, and label propagation with BLEU score

Figure 10 summarizes the impact of the weight w in our hybrid model. It shows that the model purely relies on the user-based model ($w=1.0$) achieves the best performance. Though the results suggest that the item-based model has limited performance on like prediction problem when we deal with the dataset contains only one topic, the results in Figure 7-9 show that the item-based model still has its merit.

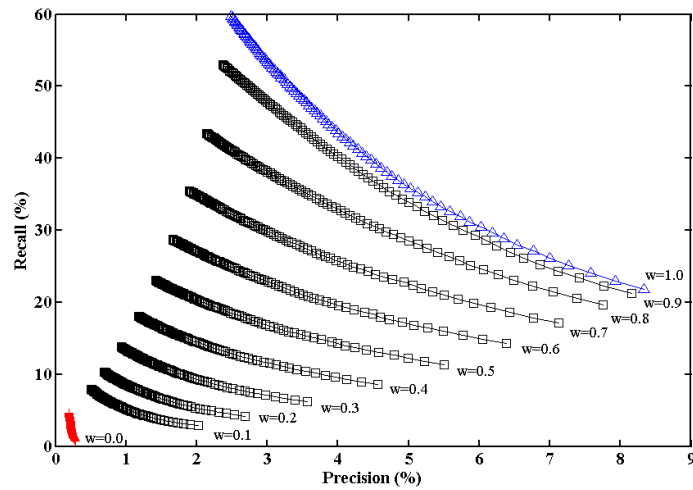


Fig. 10. ROC for different weights

6 Conclusion

In this paper, we have proposed different recommendation models based on the document content or the user engagement. The proposed models successfully utilize the similarity between documents and the probabilities from the visible units of RBM. We have shown that the proposed model outperforms the commonly adopted label propagation model. Moreover, we show that item-based model relying on semantic features cannot achieve satisfied results than user-based model. In the future, we will test more similar models and methods to integrate the user-based and item-based information to improve the probability approximation for new upcoming documents.

Acknowledgements

Research of this paper was partially supported by Ministry of Science and Technology, Taiwan, under the contract MOST 104-2221-E-001-024-MY2.

References

1. M. Kosinski, D. Stillwell, and T. Graepel, "Private traits and attributes are predictable from digital records of human behavior," *Proceedings of the National Academy of Sciences*, vol. 110, no. 15, pp. 5802-5805, 2013.
2. Y. Bachrach, M. Kosinski, T. Graepel, P. Kohli, and D. Stillwell, "Personality and patterns of Facebook usage," in *Proceedings of the 4th Annual ACM Web Science Conference*, 2012, pp. 24-32: ACM.
3. X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, p. 4, 2009.
4. G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76-80, 2003.
5. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285-295: ACM.
6. R. Van Meteren and M. Van Someren, "Using content-based filtering for recommendation," in *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, 2000, pp. 47-56.
7. I. Guy *et al.*, "Personalized recommendation of social software items based on social relations," in *Proceedings of the third ACM conference on Recommender systems*, 2009, pp. 53-60: ACM.
8. D. Carmel *et al.*, "Personalized social search based on the user's social network," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 1227-1236: ACM.
9. S. Baluja *et al.*, "Video suggestion and discovery for youtube: taking random walks through the view graph," in *Proceedings of the 17th international conference on World Wide Web*, 2008, pp. 895-904: ACM.
10. I. Guy, M. Jacovi, E. Shahar, N. Meshulam, V. Soroka, and S. Farrell, "Harvesting with SONAR: the value of aggregating social network information," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 1017-1026: ACM.
11. R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 791-798: ACM.
12. W. Chen *et al.*, "Influence Maximization in Social Networks When Negative Opinions May Emerge and Propagate," in *SDM*, 2011, vol. 11, pp. 379-390: SIAM.
13. E. Ayday, J. Zou, A. Einolghozati, and F. Fekri, "A recommender system based on belief propagation over pairwise Markov random fields," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, 2012, pp. 703-707: IEEE.
14. P. Melville, R. J. Mooney, and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," in *Eighteenth National Conference on Artificial Intelligence*, 2002, pp. 187-192: American Association for Artificial Intelligence.

15. A. B. Barragáns-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, F. A. Mikic-Fonte, and A. Peleteiro, "A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition," *Information Sciences*, vol. 180, no. 22, pp. 4290-4311, 2010.
16. K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002, pp. 311-318: Association for Computational Linguistics.
17. S. Baluja, D. Ravichandran, and D. Sivakumar, "Text Classification through Time-Efficient Label Propagation in Time-Based Graphs," in *KDIR*, 2009, pp. 174-182.
18. F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 1, pp. 55-67, 2008.