# Discriminative Factored Prior Models for Personalized Content-Based Recommendation

Lanbo Zhang, Yi Zhang
School of Engineering
UC Santa Cruz
Santa Cruz, CA, USA
{lanbo, yiz}@soe.ucsc.edu

## ABSTRACT

Most existing content-based filtering approaches including Rocchio, Language Models, SVM, Logistic Regression, Neural Networks, etc. learn user profiles independently without capturing the similarity among users. The Bayesian hierarchical models learn user profiles jointly and have the advantage of being able to borrow information from other users through a Bayesian prior. The standard Bayesian hierarchical model used in filtering assumes all user profiles are generated from the same Gaussian prior. However, considering the diversity of user interests, this assumption might not be optimal. Besides, most existing content-based filtering approaches implicitly assume that each user profile corresponds to exactly one user interest and fail to capture a user's multiple interests (information needs).

In this paper, we present a flexible Bayesian hierarchical modeling approach, which we call Discriminative Factored Prior Models (DFPM), to model both commonality and diversity among users as well as individual users' multiple interests. In our models, each user profile is modeled as a discriminative classifier with a factored model as its prior, and different factors contribute in different levels to each user profile. Compared with existing content-based filtering models, DFPM are interesting because they can 1) borrow discriminative criteria of other users while learning a particular user profile through the factored prior; 2) trade off well between diversity and commonality among users; and 3) handle the challenging classification situation where each class contains multiple concepts. We propose and implemented two specific discriminative factorization models based on different assumptions. The experimental results on a dataset collected from real digg.com users show that our models significantly outperform the baseline models of L-2 regularized logistic regression and the standard Bayesian hierarchical logistic regression models.

*CIKM'10,* October 26–30, 2010, Toronto, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0099-5/10/10 ...$10.00.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

## General Terms

Algorithms, Experimentation

## Keywords

Recommender System, Personalized Recommendation, Information Filtering, Content-based Filtering, Bayesian Hierarchical Model, Factorization

## 1. INTRODUCTION

Existing content-based filtering research is largely influenced by the Filtering Track organized by TREC, where the task is to identify documents relevant to a specific topic from a document stream. There are two major approaches to handle this task. One is to use traditional IR retrieval models (Boolean model, traditional probabilistic models, vector space model, language models, etc.) initially designed for ranking and a threshold setting algorithms for online filtering. Another approach is to treat filtering as a classification task, and thus many existing machine learning approaches (Naive Bayes, Decision Tree, Logistic Regression, SVM, Neural Networks, etc.) could be used. However, both approaches learn each user profile independently and do not make use of the commonality among users.

Yu et al[4] and Zhang et al[6] introduced the Bayesian hierarchical modeling approach to ***jointly*** learn user profiles for content-based filtering. Based on the fact that different users may have similar interests, the Bayesian hierarchical models assume that all user profiles are sampled from a common Gaussian prior. The Bayesian hierarchical modeling approach helps alleviate the cold-start problem since it is able to borrow discriminative information from other users through the common prior when learning a particular user profile, especially for those users with little training data.

However, some users may have totally different interests, and requiring these users' profiles to follow the same Gaussian prior distribution may negatively influence the learned profiles, thus we need to trade off better between the diversity and commonality of user profiles. Besides, almost all existing content-based filtering approaches cannot capture the multiple interests of a user. They implicitly assume each user profile only corresponds to a single interest, which does not fit the real scenarios in personalized recommendation, where a real user's interests may contain multiple con-

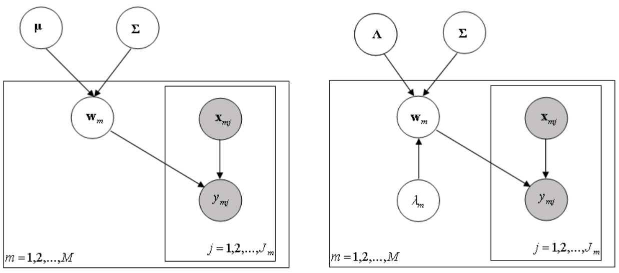


**Figure 1: Left: The Bayeisn Hierarchical Model (BHM). Right: The Discriminative Factored Prior Model.**

cepts/topics. For example, a graduate student working on information retrieval may be interested in both IR research advancements and NBA news.

To better model the diversity and commonality of users and each user's multiple interests, this paper proposes a flexible Bayesian hierarchical modeling approach for personalized content-based recommendation.

The discriminative factorization models in this paper are motivated by the theoretical attractiveness of factorized topic models as well as the empirical success of discriminative models. Factorization-based generative topic models, such as Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA) are very attractive theoretically. However, we are not aware of any empirical evidence of their effectiveness in competitive retrieval and filtering tasks. This is not surprising since most of the successful models in these tasks are discriminative models, such as Logistic Regression or SVM, instead of generative models, such as Naive Bays. Though our models also introduce latent factors, there are clear differences between our models and PLSA/LDA. Unlike PLSA and LDA, we learn user profiles as discriminative functions. The entries of each factor in our models are not necessarily words, and could be any item features such as the time or authority of a document. We noticed that our model is very similar to the multi-task learning model proposed by an independent group of researchers [5]. However, the existing paper was focusing on multi-task learning and used Reuters document classification task for evaluation, while our work emphasizes on the application of personalized recommendation, which has very different characteristics and challenges.

## 2. DISCRIMINATIVE FACTORED PRIOR MODELS (DFPM)

Our Discriminative Factored Prior Models (Figure 1 (right)) are motivated by the well known Bayesian hierarchical model in Figure 1 (left)[4, 6]. There are $M$ users in total and each user has $J_m$ training examples. $\mathbf{w}_m$ is the profile of user $m$. In BHM, all user profiles follow a Gaussian prior distribution with mean vector $\mu$ and covariance matrix $\mathbf{\Sigma}$. In DFPM, $\mathbf{\Lambda}$ is a $K \times H$ matrix, and $\lambda_\mathbf{m}$ is a user-dependent vector with length $H$. The product of $\mathbf{\Lambda}$ and $\lambda_m$ determines the prior mean of each user profile $\mathbf{w}_m$, and $\mathbf{\Sigma}$ is the prior covariance matrix. The assumption of DFPM is: there are a number of hidden factors that represent different decision boundaries in the item feature space; users may be using one or several of these hidden factors in different levels. Each column of matrix $\mathbf{\Lambda}$, which is a $K$-dimendional vector, represents a specific hidden factor. $\lambda_m$ tells how much each column of $\mathbf{\Lambda}$ contributes to the profile of user $m$.

$\lambda_m$ may follow two alternative distributions: Multinomial and Normal, and we will use **DFPM-Mult** and **DFPM-Norm** to denote these two models respectively. In DFPM-Mult, only one entry of $\lambda_m$ is allowed to be 1 and all other entries be zero. This model clusters users into $H$ groups, and users of the same group share a common hidden factor as the prior. We want to point out that the common Bayesian hierarchical model (BHM) is actually a special case of DFPM-Mult. When $H = 1$, DFPM-Mult is equivalent to BHM. In the case of DFPM-Norm, $\lambda_m$ follows a Normal distribution with mean $\mathbf{0}$ and variance $b\mathbf{I}$. DFPM-Norm assumes that each user may be interested in multiple hidden factors, and each entry of $\lambda_m$ reflects how much the corresponding hidden factor is related to the user's interests. DFPM-Norm is used to capture each user's multiple interests.

We assume each user profile $\mathbf{w}_m$ is a random sample from a normal distribution with mean $\mathbf{\Lambda}\lambda_m$ and variance $\mathbf{\Sigma}$. The label $y_{m_j}$ of a training item $\mathbf{x}_{m_j}$ is $y = f(\mathbf{x}_{m_j}; \mathbf{w}_m)$, where $f$ could be many existing regression or classification models. We will take the logistic regression as an example to demonstrate how our models could be used for recommendation task. Let $\mathbf{I}$ be an identity matrix, $a, b, c$ be constant parameters, we summarize the discriminative factored prior models with logistic regression as follows:

- Each column of hidden factor matrix $\mathbf{\Lambda}$ follows a normal distribution: $N(\mathbf{0}, \mathbf{aI})$. $\mathbf{\Sigma}$ follows an Inverse Wishart distribution $\mathbf{W^{-1}(I, c)}$
- The user vector $\lambda_m$ may follow two alternative distributions: Normal or Multinomial. In the case of DFPM-Norm, $\lambda_m \sim N(\mathbf{0}, \mathbf{bI})$; and in the case of DFPM-Mult, $\lambda_m \sim Multinomial(\frac{1}{H}, \frac{1}{H}, ..., \frac{1}{H})$
- The user profile $\mathbf{w}_m$ follows a normal distribution: $\mathbf{w}_m \sim N(\mathbf{\Lambda}\lambda_m, \mathbf{\Sigma})$
- Given a user profile $\mathbf{w}_m$ and an item feature vector $\mathbf{x}_{mj}$, its label is sampled from:

$$y_{mj} \sim Bernoulli(\frac{1}{1 + exp(-\mathbf{w}_m^T \mathbf{x}_{mj})})$$

**Parameter Estimation:** The empirical Bayes method[2] is often used when learning a complicated Bayesian hierarchical model like DFPM. However, the learning algorithm based on Empirical Bayes will be very complicated since there are many hidden variables ($\mathbf{\Lambda}$, $\mathbf{\Sigma}$, $\lambda$, $\mathbf{w}_m$) entangled in our models. Besides, the number of features and the number of users involved in the recommendation task are usually huge. Thus the empirical Bayes method is too computationally expensive to be used here. As an alternative, we use a simplified learning algorithm based on point estimation and the conjugate gradient decent algorithm. For those who are interested in the algorithm, please refer to the longer version of this paper at http://users.soe.ucsc.edu/∼lanbo/cikm10long.pdf.

## 3. EXPERIMENTAL METHODOLOGY

To evaluate the proposed modeling approach, we collected a data set from Digg.com[1]. Digg.com is a website for people to share web content including news, images, and videos. Users can digg items they are interested in to promote the

Table 1: Statistics of the crawled Digg data

| | |
|---|---|
| Number of users | 15,162 |
| Number of news articles | 91,088 |
| Total number of diggs | 3,809,196 |
| Average number of diggs per user | 251 |

items' ranking. Each item dugg by a user is considered a positive data point (a relevant document) for the user. We collected the complete digg history of news articles of more than 15,000 users. The detailed statistics of our dataset is shown in Table 1.

Since Digg.com only has user digg history available on its website, we couldn't get those articles users read but didn't digg. In other words, we don't have real negative examples. To address this problem, we randomly choose equal number of articles which are not dugg by a user as the negative examples for this user. Considering the large percentage of user undugg articles, we expect most of the articles sampled in this way are irrelevant to this user's interests.

We randomly divide each user's data (including both positive and negative examples) into three parts: training (80%), validation (10%), and test (10%). The validation data is used to tune the parameters of both our models ($H, c_1, c_2, c_3$) and the baseline models. We use the words as features. Both the stop words and rare words (occurring in less than 50 articles) are removed from the feature set. As a result, there are 35,865 features. When calculating the feature values, we use the TF*IDF scoring method. **Precision**, **Recall**, and **Macro-F1** are used as the evaluation measures.

Our experiments are designed to answer the following questions: 1) How is the performance of our models compared with the state-of-the-art algorithms? 2) Can our models learn meaningful hidden factors, and how does the number of hidden factors ($H$) influence the performance? To answer the first question, we compare our models with the L-2 regularized logistic regression (**L2LR**) and the Bayesian hierarchical model (Figure 1 (left)) with logistic regression (**BHLR**) implemented in [3], since other researchers have demonstrated that BHLR works much better than popular generative filtering models [4, 6]. To answer the second question, the results with different numbers of hidden factors will be compared and analyzed.

# 4. EXPERIMENTAL RESULTS

## 4.1 Overall Performances

The top-left graph in Figure 2 shows the overall performances of four algorithms. Both of our models (DFPM-Norm and DFPM-Mult) are statistically ***significantly*** better than the baselines in terms of Precision and Macro-F1 (based on t-test). The improvement on precision is very significant. This is very encouraging since Precision is a more important factor in most personalized recommender/filtering systems. To see whether our algorithms are helpful for both hard users (users with little training data) and easy users (users with much training data), we divide the users into five groups according to their numbers of diggs (less than 50, 50-100, 100-200, 200-500, greater than 500 respectively) to see whether the performances for all kinds of users have been improved. The rest graphs in Figure 2 show the results on these user groups. We find our models outperform the baselines for all five user groups.

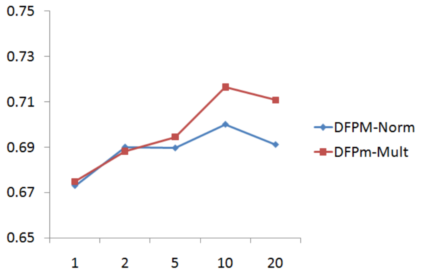


**Figure 3: Performances (Macro-F1) of our models at different $H$ (number of hidden factors)**

## 4.2 DFPM v.s. Baselines

Figure 2 shows that our models ***significantly*** outperform the L-2 regularized logistic regression, which learns each user profile independently. This demonstrates that our models successfully borrow discriminative information from other users by learning user profiles jointly. Figure 2 also shows that our models ***significantly*** outperform the Bayesian hierarchical logistic regression model (BHLR). We find that BHLR already significantly outperforms L2LR, which indicates BHLR successfully borrows information from other users. Encouragingly, our models can further improve the performances over BHLR. This demonstrates that our models can learn more accurate user profiles by introducing a factored prior.

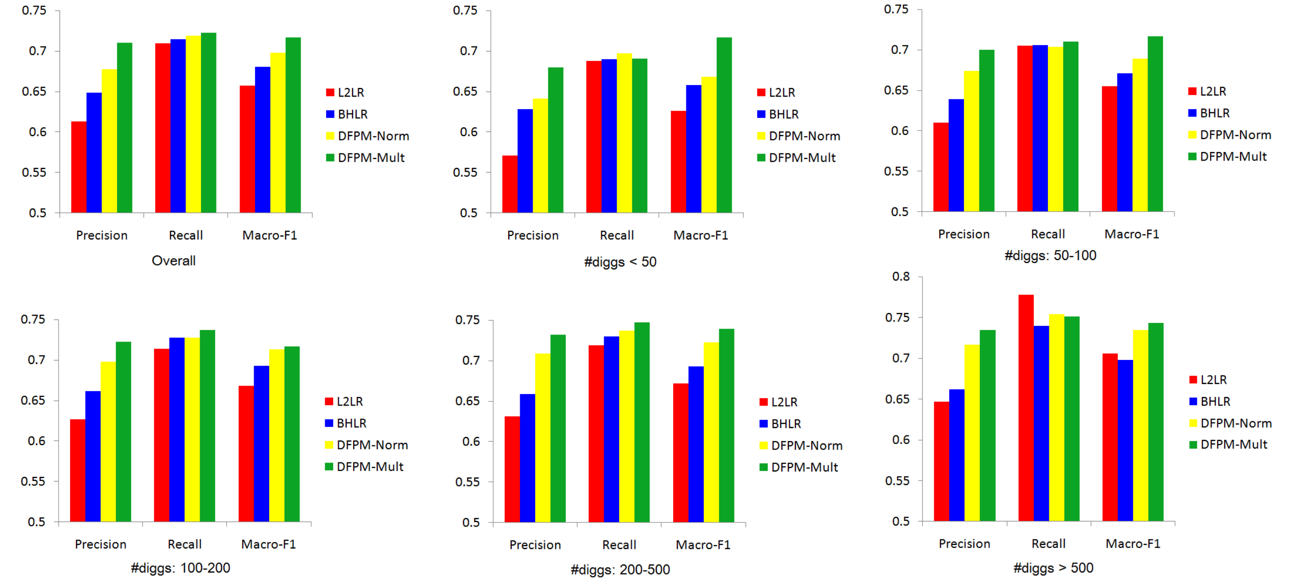
Why our models can outperform BHLR? Not all users have similar interests, and it's not always a good idea to assume that all user profiles are generated from the same Gaussian distribution. Our models have less strong assumptions and use the variable $\lambda_m$ to model the diversity of users, and thus have the advantage of only borrowing information from ***similar*** users. In particular, users with similar interests share a common hidden factor as the prior in the DFPM-Mult model.

## 4.3 The Hidden Factors

Figure 3 shows how the number of hidden factors influences the performance. Remember BHLR is a special case ($H = 1$) of our model DFPM-Mult. As $H$ increases from 1 to 10, the performance keeps on improving and reaches the optimal value at $H = 10$. If we consider users with similar interests as a cluster, our model DFPM-Mult can effectively identify the underlying user clusters. To better understand the DFPM-Mult model, we list the top 10 features (words) in some learned factors in Table 2. We observe that most of the words represent the concept of each hidden factor well.

## 4.4 $\lambda_m$: Multinomial v.s. Normal

Figure 2 shows that DFPM-Mult performs better than DFPM-Norm. This is somewhat surprising. Initially, we expected that DFPM-Norm should perform better than DFPM-Mult since the Normal assumption of $\lambda_m$ can capture the multiple interests of individual users. There are several possible reasons for this finding. First, we probably should learn a prior for $\lambda_m$ instead of using a normal distribution centered on 0. Second, it's possible that the flexibility of $\lambda_m$



Overall

#diggs < 50

#diggs: 50-100

#diggs: 100-200

#diggs: 200-500

#diggs > 500

**Figure 2: Performance comparison of different models. L2LR: L-2 regularized logistic regression. BHLR: Bayesian hierarchical logistic regression model. DFPM-Norm and DFPM-Mult are our models. The top-left graph shows the overall performance on all users, and the rest on five user groups with different #diggs.**

**Table 2: Top words in some factors (by model DFPM-Mult).**

| | | | |
|---|---|---|---|
| obama | scientist | smartphon | linux |
| presid | relationship | tablet | mozilla |
| jed | mlm | chrome | chrome |
| palin | exercis | android | diggtv |
| beck | geograph | dropbox | broadband |
| marijuana | treatment | feb | anonym |
| mccain | coach | broadband | techradar |
| yellow | copenhagen | diggtv | lifehack |
| barack | foreclosur | dialogg | dialogg |
| religi | orbit | chines | interfac |

makes the learning process more complicated. It may curtail the information borrowed from other users so that the commonality of similar users is not captured well. We are planning to investigate the reasons in more details in the future work.

## 5. CONCLUSIONS

In this paper, we present the discriminative factored prior models for personalized content-based recommendation. Particularly, we propose two models and the corresponding parameter learning algorithms. We evaluate our models on a dataset collected from real web users on Digg.com[1], and compare them with two much related baseline algorithms. The experimental results demonstrate that: 1) Our models significantly improve the recommendation performance, especially for users with little training data. Thus they can help alleviate the cold-start problem. 2) It's helpful to introduce a factorized prior. Particularly, the DFPM-Mult model learns more accurate user profiles since it can effectively cluster users with similar interests and has the advantage of only borrowing discriminative criteria from similar users while learning a particular user profile.

In the future work, more research is needed to analyze DFPM-Norm, since it captures each user's multiple interests and thus offers some advantages over the DFPM-Mult model. One possible approach is to try a learned prior or using a Dirichlet prior instead of a normal prior for $\lambda_m$. Besides, we will also evaluate our models on more datasets. Our models can also be modified to fit the personalized recommendation task better, for example, to capture user interest drift by adding temporal variables.

## 6. ACKNOWLEDGMENTS

This work was funded by National Science Foundation IIS-0713111, AFRL/AFOSR and UCSC/LANL ISSDM. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsors.

## 7. REFERENCES

[1] The digg website. https://www.digg.com.

[2] Empirical bayes method. http://en.wikipedia.org/wiki/Empirical_Bayes_method.

[3] Hierarchical modeling in bbr. http://www.stat.rutgers.edu/∼madigan/BBR/hier.html.

[4] K. Yu, V. Tresp, and S. Yu. A nonparametric hierarchical bayesian framework for information filtering. In *SIGIR*, 2004.

[5] J. Zhang, Z. Ghahramani, and Y. Yang. Flexible latent variable models for multi-task learning. *Mach. Learn.*, 73(3):221–242, 2008.

[6] Y. Zhang and J. Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *SIGIR '07*, pages 47–54, New York, NY, USA, 2007. ACM.
]