# Divide and Transfer: Understanding Latent Factors for Recommendation Tasks

Vidyadhar Rao
TCS Research Labs, India
vidyadhar.rao@tcs.com

Rosni K V[*]
University of Hyderabad, India
rosnikv@gmail.com

Vineet Padmanabhan
University of Hyderabad, India
vineetcs@uohyd.ernet.in

## ABSTRACT

Traditionally, latent factor models have been the most successful techniques to build recommendation systems. While the key is to capture the user interests effectively, most research is focused on learning latent factors under cold-start and data sparsity situations. Our work brings a complementary approach to the previous studies showing that understanding the semantic aspects of latent factors could give a hint on how to transfer useful knowledge from auxiliary domain(s) to the target domain.

In this work, we propose a collaborative filtering technique that can effectively utilize the user preferences and content information. In our approach, we follow a divide and transfer strategy that could derive semantically meaningful latent factors and utilize only the appropriate components for recommendations. We demonstrate the effectiveness of our approach due to improved latent feature space in both single and cross-domain tasks. Further, we also show its robustness by performing extensive experiments under cold-start and data sparsity contexts.

## CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Collaborative filtering**; • **Computing methodologies** → **Topic modeling**; **Learning latent representations**;

## 1 INTRODUCTION

Most of the e-commerce businesses would want to help the customers surf through items that might interest them. Some examples include recommending books by Goodreads, products by Amazon, movies by Netflix, music by Last.fm, news articles by Google, etc. The most popular recommender systems follow two paradigms: *Collaborative filtering (CF):* utilize the preferences from a group of users and suggest items to other users; and *Content-based filtering:* recommend items that are similar to those that a user liked in the past. In general, effective recommendations are obtained by combining both content-based and collaborative features.

While many of these approaches are shown to be effective (in single domain), but in practice they have to operate in challenging environment (in cross domain), and deliver more desirable recommendations. For example, based on user's watched list on movies, it may be easy to recommend upcoming new movies, but how do we recommend books that have similar plots. Typically, in single-domain user preferences from only one domain are used to recommend items within the same domain, and in cross-domain user preferences from auxiliary domain(s) are used to recommend items on another domain. *Hence, producing meaningful recommendations depends on how well the assumptions on the source domain align with the operating environment.* While the key is to transfer user interests from source to target domain, this problem has two characteristics: (1) *Cold-start problem* i.e., shortage of information for new users or new items; and (2) *Data sparsity problem* i.e., users generally rate only a limited number of items.

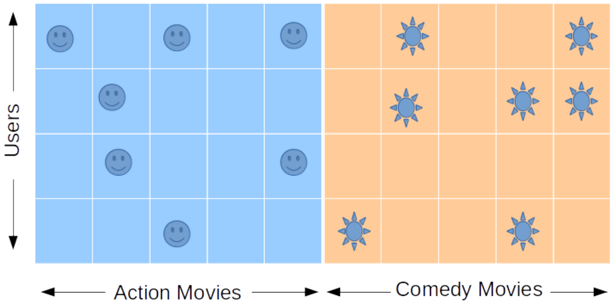


**Figure 1: Illustration of cross domain scenario: Same users might have different rating preferences for two genres in movie recommendation system.**

Traditionally, in single-domain, the latent factor models [8] are used to transform the users and items into a common latent feature space. Intuitively, users' factors encode the 'preferences' while the item factors encode the 'properties'. However, the user and item latent factors have no interpretable meaning in natural language. Moreover, these techniques fail in the cross-domain scenarios because the learned latent features may not align over different domains. *Thus, understanding the semantic aspects of the latent factors is highly desirable in cross domain research under cold-start and data sparsity contexts.*

In cross domain scenario, tensor factorization models [7] try to represent the user-item-domain interaction with a tensor of order three and factorize users, items and domains into latent feature vectors. Essentially, these methods improve recommendations when the rating matrices from auxiliary domains share similar user-item rating patterns. However, the user behavior in all domains may not always be same and each user might have different domains of interest (see Fig. 1). Moreover, when the auxiliary information from multiple sources are combined, learned latent features may some times degrade the quality of recommendations. *(see Section 6.2.1)*

[*] The work was conducted during an internship at TCS Research Labs, India.

*RecSysKTL Workshop @ ACM RecSys '17, August 27, 2017, Como, Italy*

To address these shortcomings, we propose a method that can derive semantically meaningful latent factors in a fully automatic manner, and can hopefully improve the quality of recommendations. The major contributions of this work are:

(1) We hypothesize that the intent of the users are not significantly different with respect to the document-specific and corpus-specific background information and thus, they can be ignored when learning the latent factors.
(2) We propose a collaborative filtering technique that segments the latent factors into semantic units and transfer only useful components to target domain. *(see Section 4)*
(3) We demonstrate the superiority of the proposed method in both single and cross-domain settings. Further, we show consistency of our approach due to improved latent features under the cold-start and data sparsity contexts. *(see Section 6)*

## 2 RELATED WORK

Among the latent factor models, matrix factorization (and it's variants) [8] is the popular technique that tries to approximate an observed rating matrix to derive latent features. The basic principle is to find a common low-dimensional representation for both users and items i.e, reduce the rank of user-item matrix directly. Nevertheless, the reduction approach addresses the sparsity problem by removing the unrepresentative or insignificant users or items. Discarding any useful information in this process may hinder further progress in this direction. To mitigate the cold start problem, their variants [6] exploit user preferences or behavior from implicit feedbacks to improve personalized recommendations. Our work differs from these as we do not completely remove the content information. Instead, we assume that the content information about the items could be captured from multiple semantic aspects, which are good at explaining the factors that contributed to the user preferences.

Another popular latent factor model is one-class collaborative filtering [14] that tries to resolve the data sparsity problem by interpreting the missing ratings as a mixture of negative examples and unlabeled positive examples. Essentially, their task is to distinguish between user's lack of interest in an item to user's lack of awareness of the item. Alternately, others exploit the user generated information [9, 19] or create a shared knowledge from rating matrices in multiple domains [4, 13, 15, 16]. However, they have limited utility as users tend to show different ratings patterns across the domains. In our work, we do not use the user generated content, and we only use the user preferences along with the content information of items to learn the latent feature space.

While all these approaches try to mitigate the cold-start and data sparsity in the source domain, we focus on understanding the semantics aspects of the learned latent factors. Many methods tried [11, 18, 21, 23] to adjust the latent factors according to the context. For example, in the job recommendation application, the cross-domain segmented model [18] introduces user-based domains to derive indicator features and segment the users into different domains. A projection based method [23] learns a projection matrix for each user that is able to capture the complexities of their preferences towards certain items over others. Our work differs from these in the sense that we consider the domains based on item features and exploit the contextual information (such as text) in order to understand the meaning of the latent factors.

In our research, we built an algorithm inspired by technique[1] collaborative topic modeling [22] that can make recommendations by adjusting user preferences and content information. We combine this model with the specific word with background model [3] that can account for the semantic aspects of texts from multiple domains. Thus, our model can transfer only the useful information to the target domain by improving the latent feature space. To validate our hypothesis, we conducted experiments in both single and cross domain recommendations in extreme cold-start and data sparsity scenarios, and reflect on the factors effecting the quality of recommendations.

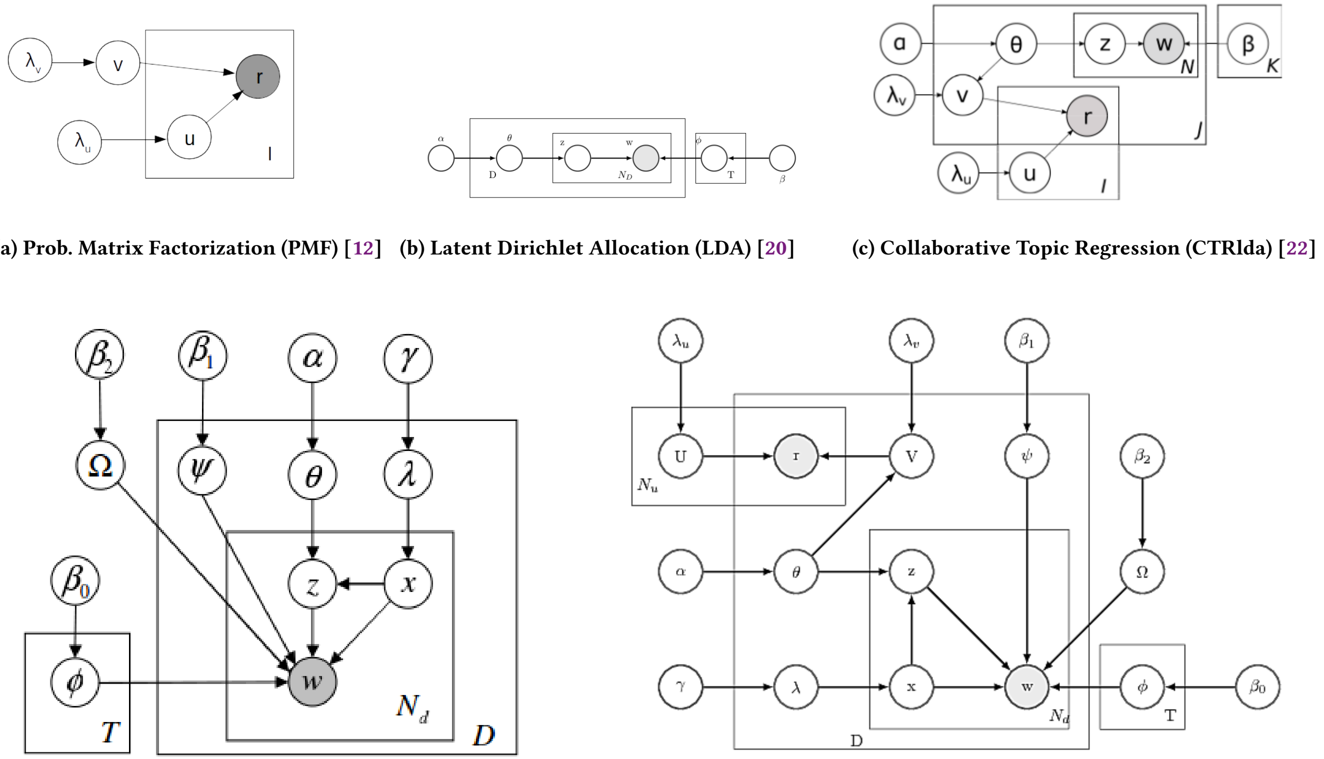
**Table 1: Notations used in this paper**

| | |
|---|---|
| User node | $U$ |
| Item node | $V$ |
| Rating node | $r$ |
| Number of users | $N_u$ |
| Regularization parameter for user | $\lambda_u$ |
| Regularization parameter for item | $\lambda_v$ |
| Number of documents(items) | $D$ |
| Number of topics | $T$ |
| Number of words in a document ($d$) | $N_d$ |
| $n^{th}$ word in the document ($d$) | $w_{dn}$ |
| Topic assignment for document ($d$) and word ($w_{dn}$) | $z_d$,$z_{dn}$ |
| Switch variable | $x$ |
| Prob. of topics given documents ($T \times D$) | $\theta$ |
| Prob. of switch variable given document ($D \times 3$) | $\lambda$ |
| Prob. of words given special word distribution of document ($d$) | $\psi_d$ |
| Prob. of words given topics ($W \times T$) | $\phi$ |
| Prob. of words given background distribution | $\Omega$ |
| Dirichlet prior on document-topic distribution, $\theta$ ($D \times T$) | $\alpha$ |
| Dirichlet prior on switch variable distributions | $\gamma$ |
| Dirichlet prior on word-topic distribution ($\phi$) | $\beta_0$ |
| Dirichlet prior on special word distribution ($\psi$) | $\beta_1$ |
| Dirichlet prior on background word distribution ($\Omega$) | $\beta_2$ |

## 3 PRELIMINARIES

Our method follows the same line as the collaborative topic regression model (CTR) [22], in the sense that latent factors of the content information are integrated with the user preferences. The main difference of our model with this approach is the way in which we derive meaningful topical latent factors from the content information and enable better predictions on recommendation tasks in general. For this we use the specific word with background model [3]. Before we describe our model, we give a brief review of the existing latent factor models which serve as a basis for our approach. The notations for graphical models are given in Table 1.

[1] https://open.blogs.nytimes.com/2015/08/11/building-the-next-new-york-times-recommendation-engine/



**(a) Prob. Matrix Factorization (PMF) [12]** **(b) Latent Dirichlet Allocation (LDA) [20]** **(c) Collaborative Topic Regression (CTRlda) [22]**

**(d) Special Word with Background (SWB) [3]**

**(e) Proposed Divide and Transfer Model (CTRswb)**

**Figure 2: Graphical representation of the latent topic models and recommendation models: (a) Probabilistic Matrix Factorization (PMF), (b) Latent Dirichlet Allocation model (LDA), (c) Special words with background model (SWB), (d) Collaborative Topic Regression (CTRlda), (e) Proposed divide and transfer model (CTRswb). The hidden nodes —the topic proportions, assignments, and topics —are unshaded. The observed nodes —the ratings and words of the documents —are shaded.**

### 3.1 Probabilistic Matrix Factorization

Matrix factorization models the user-item preference matrix as a product of two lower-rank user and item matrices. Given an observed matrix, the matrix factorization for collaborative filtering can be generalized as a probabilistic model (PMF) [12], which scales linearly with the number of observations. The graphical model for PMF is shown in Fig. 2a. In the figure, a user $i$ is represented by a latent vector $u_i \in R^K$ and an item $j$ by a latent vector $v_j \in R^K$, where $K$ is the shared latent low-dimensional space.

### 3.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [1, 20] is known to be powerful technique for discovering and exploiting the hidden thematic structure in large archives of text. The principle behind LDA is that documents exhibit multiple topics and, each topic can be viewed as a probability distribution over a fixed vocabulary. The richer structure in the latent topic space allows to interpret documents in low-dimensional representations. Fig. 2b depicts the graphic model for the LDA model where the latent factors word-topic ($\phi$) and the topic-document ($\theta$) are inferred from a given collection of documents.

### 3.3 Collaborative Topic Regression

Collaborative topic regression (CTRlda) model [22] combines the latent topics from LDA, and the user-item features from PMF to jointly explain the observed content and user ratings, respectively. This model has two benefits over traditional approaches: *(a) generalization to unseen or new items (b) generate interpretable user profiles*. The graphic model for CTRlda model is shown in Fig. 2c. More details about this model are given in section 4.

### 3.4 Special Word Topic Model

Special words with background model (SWB) [3] is based on LDA which models words in a document as either originating from general topics, or from document-specific word distributions, or from a corpus-wide background distribution. To achieve this, the SWB model introduces additional switch variables into the LDA model to account for multiple word distributions. The SWB model has similar general structure to the LDA model as shown in Fig. 2d. The main advantage of SWB model is that it can trade-off between generality and specificity of documents in a fully probabilistic and automated manner. An incremental version of this model exploits this feature to build an automatic term extractor [10].

# 4 DIVIDE AND TRANSFER LATENT TOPICS

Consider we have a set of $I$ users, $J$ items, and the rating variable $r_{ij} \in \{0, 1\}$ that indicates if user $i$ likes item $j$ or not. For each user, we want to recommend items that are previously unseen and potentially interesting. Traditionally, the latent factor models try to learn a common latent space of the users and items, given an observed user-item preference matrix. Essentially, the recommendation problem minimizes the regularized squared error loss with respect to the $(u_i)_{i=1}^{I}$ and $(v_j)_{j=1}^{J}$,

$$min \sum_{i,j} (r_{ij} - u_i^T v_j)^2 + \lambda_u ||u_i||^2 + \lambda_v ||v_j||^2 \qquad (1)$$

where $\lambda_u$ and $\lambda_v$ are regularization parameters. Probabilistic matrix factorization (PMF) [12] solves this problem by drawing the ratings for a given user-item pair from a Gaussian distribution given by

$$\hat{r_{ij}} \sim \mathcal{N}(u_i^T v_j, c_{ij}) \qquad (2)$$

where $c_{ij}$ is a confidence parameter for $r_{ij}$. In our work, we are interested in jointly modeling the user preferences and the content information to improve the quality of recommendations. We strict to the assumption that the content information from single/multiple domain(s) and users share a common latent topic space. Our model builds on the CTRlda [22] model which can effectively balance the user preferences and the content information. This is achieved by including the latent variable $\epsilon_j$ that offsets the topic proportion $\theta_j$ i.e., $v_j = \theta_j + \epsilon_j$, where the item latent vector $v_j$ is close to the topic proportion $\theta_j$ derived from LDA and could diverge from it if it has to. Here, the expectation of rating for a user-item pair is a simple linear function of $\theta_j$, i.e.,

$$E[r_{ij} \mid u_i, \theta_j, \epsilon_j] = u_i^T (\theta_j + \epsilon_j) \qquad (3)$$

This explains how much of the prediction relies on content and how much it relies on how many users have rated an item. We propose a straightforward extension of CTRlda that replaces the topic proportions derived from LDA [1] with multiple semantic proportions derived from SWB [3] over the common topic space.

## 4.1 Graphical Model

Our model is based on placing additional latent variables into CTRlda model that can account for semantic aspects of the latent factors. The graphical model of divide and transfer model, referred as 'CTRswb', is shown in Fig. 2e.

### 4.1.1 *Deriving Semantic Factors.*

[3] As can be seen in the figure, the latent variable $x$, associated with each word, acts as switch: when $x = 0$, the word is generated via topic route; when $x = 1$, it is generated via document-specific route; and for $x = 2$, it is generated via background route which is corpus specific. For $x = 0$ case, like LDA, words are sampled from document-topic ($\theta$) and word-topic ($\phi$) multinomials with $\alpha$ and $\beta_0$ as respective symmetric Dirichlet priors. For $x = 1$ or $x = 2$, words are sampled from document-specific ($\psi$) or corpus-specific ($\Omega$) multinomials with $\beta_1$ and $\beta_2$ as symmetric Dirichlet priors, respectively. The variable $x$ is sampled from a document-specific multinomial $\lambda$, which in turn has a symmetric Dirichlet prior, $\gamma$. Since, the words are sampled from mutiple topic routes, our model can automatically deduce the latent features in a precise and meaningful manner.

## 4.2 Posterior Inference

The generative process is described in Algorithm 1 which combines SWB [3] model (lines 1-14) and PMF [12] model (lines 15-26). We summarize the repeated sampling of word distributions for each topic and user factors, and the predictions of user-item ratings.

### 4.2.1 *Learning Parameters.*

Computing full posterior of the parameters $u_i, v_j, \theta_j$ is intractable. Therefore, we adapt the EM-style algorithm, as in CTRlda [22], to learn the maximum-a-posteriori estimates. We refer the interested reader to CTRlda [22] for more details. It was shown that fixing $\theta_j$ as the estimate gives comparable performance with vanilla LDA. We discovered that EM algorithm convergence improved significantly when $\theta_j$ from the SWB topic route is used as initial estimate. (see Fig. 3b)

### 4.2.2 *Making Predictions.*

After learning (locally) all the parameters, subject to a convergence criteria, we can use the learned latent features $(u_i)_{i=1}^{I}$, $(v_j)_{j=1}^{J}$, $(\theta_j^*)_{j=1}^{J}$, $(\epsilon_j^*)_{j=1}^{J}$ in Eq. (3) for predictions. Note that for new or unseen items, we do not have the offset value i.e., $\epsilon_j = 0$, hence the prediction completely relies on the topic proportion derived from either latent models LDA or SWB model.

### 4.2.3 *Discussion.*

It is a common practice that the ratings of items are given on a scale, and the latent models try to predict the rating for a new user-item pair. In such cases, the factorization machines [17] are known to work for a variety of general prediction tasks like classification, regression, or ranking. In our setup, the ratings are binary i.e., $r_{ij} \in \{0, 1\}$ where $r_{ij} = 0$ can be interpreted in two ways: *either $u_i$ is not interested in $v_j$, or $u_i$ does not know about $v_j$*. In a way our goals differ from the prediction tasks considered in factorization machines. Our study shows we can make predictions to unseen items while deriving meaningful latent factors.

While making predictions to unseen items, it is important to see how effectively they can fuse the content information from multiple sources. In our model, the semantic units are effective for representation of latent factors, and has advantages over CTRlda model. While the user preferences across the domains are very different (see Fig. 1), the background word distributions are nearly similar across all items, and therefore, its contribution towards $v_j$ is not significant. Additionally, the specific words that occur in the documents do not convey much information about the user preferences. Hence, we can discard the $\Omega, \psi$ distributions and only use the $\theta_j$ derived from the general topic route of the SWB [3] model. Subsequently, we demonstrate that CTRswb could learn better representations for the latent features compared to the CTRlda [22].

# 5 EXPERIMENTS

We demonstrate the efficacy of our approach (CTRswb) in both single and cross domain scenarios on CiteULike dataset and MovieLens dataset, respectively. For single domain, we adapt the same experiment settings as that of CTRlda [22]. Since, cross-domain applications can be realized in multiple ways [2], we consider the shared user's setup across multiple domains in two different contexts: (1) recommendations in cold-start context, where we study the impact of number of topics in the auxiliary domain(s) and (2) recommendations in data sparsity context, where we study the impact of number of ratings in the auxiliary domain(s).

**Algorithm 1:** Generative process for CTRswb model

```
Select a background distribution over words $\Omega|\beta_2 \sim Dir(\beta_2)$
for each topic $k \in 1, ...., T$ do
    Select a word distribution $\phi_k|\beta_0 \sim Dir(\beta_0)$
end
for each document $d \in 1, ...D$ do
    Select a distribution over topics $\theta_d|\alpha \sim Dir(\alpha)$
    Select a special-words distribution over words $\psi_d|\beta_1 \sim Dir(\beta_1)$
    Select a distribution over switch variables $\lambda_d|\gamma \sim Beta(\gamma)$
    for $n = 1 : N_d$ words in document $d$ do
        Select a switch variable $x_{dn}|\lambda_d \sim Mult(\lambda_d)$
        Select $z_{dn}|\{\theta_d, x_{dn}\} \sim Mult(\theta_d)^{\delta(x_{dn},1)}\delta(z_{dn}, SW)^{\delta(x_{dn},2)}\delta(z_{dn}, BG)^{\delta(x_{dn},3)}$
        Generate a word: $w_{dn}|\{z_{dn}, x_{dn}, \phi, \psi_d, \Omega\} \sim Mult(\phi_{z_{dn}})^{\delta(x_{dn},1)}Mult(\psi_d)^{\delta(x_{dn},2)}Mult(\Omega)^{\delta(x_{dn},3)}$
    end
end
for user $i \in 1...N_u$ do
    Draw $u_i \sim \mathcal{N}\ (0, \lambda_u^{-1}\ \mathcal{I}_T)$
end
for item $j \in 1...D$ do
    Draw $\epsilon_j \sim \mathcal{N}\ (0, \lambda_v^{-1}\ \mathcal{I}_T)$
    Compute $v_j = \epsilon_j + \theta_j$
end
for user $i \in 1...N_u$ do
    for item $j \in 1...D$ do
        Draw r$_{ij} \sim \mathcal{N}\ (u_i^T v_j, c_{ij})$
    end
end
```

## 5.1 CiteULike Dataset

We conducted experiments in single domain using dataset from the CiteULike [2], a free service social network for scholars which allows users to organize (personal libraries) and share papers they are reading. We use the metadata of CiteULike from [22] collected during 2004 and 2010. The dataset contains 204, 986 pairs of observed ratings with 5551 users and 16, 980 articles. Each user has 37 articles in their library on an average and only 7% of the users has more than 100 articles. That is, the density of dataset is quite low: 0.2175%. Item or article is represented by it's title and abstract. After pre processing the corpus, 8000 unique words are generated as vocabulary.

## 5.2 MovieLens Dataset

To conduct the experiments in cross-domain, we have used the dataset provided by Grouplens [5]. We extracted five genres with most ratings out of the 19 genres from the 1 million movielens dataset: *Action, Comedy, Drama, Romance, Thriller.* Since the movielens dataset has only user generated tags, we crawled the IMDB [3] to get item description for the movies. The basic statistics of the dataset collected are reported in the Table 2.

**Table 2: MovieLens 1M Data statistics**

| Movie Genre | No.Items | No.Users | No.Ratings | Rating Ratio |
|---|---|---|---|---|
| Drama | 1,493 | 5,881 | 352,834 | 0.040 |
| Comedy | 1,163 | 5,881 | 354,455 | 0.052 |
| Thriller | 485 | 5,881 | 188,968 | 0.066 |
| Romance | 459 | 5,881 | 146,916 | 0.054 |
| Action | 495 | 5,881 | 256,515 | 0.088 |
| Total | 4,095 | 5,881 | 1,299,688 | 0.054 |

## 5.3 Evaluation Methodology and Metrics

We evaluate the recommendation tasks by using the standard performance metrics: Precision, Recall and Mean Average Precision(MAP). The results shown are averaged over all the users. In our studies, we set the parameters of PMF and CTRlda by referring to [22]. For PMF, $\lambda_u = \lambda_v = 0.01$, $a = 1$, $b = 0.01$. For CTRlda model, $T = 200$, $\lambda_u = 0.01$, $\lambda_v = 100$, $a = 1$, $b = 0.01$. For CTRswb model, we set $\alpha = 0.1$, $\beta_0, \beta_2 = 0.01$, $\beta_1 = 0.0001$, $\gamma = 0.3$ (all weak symmetric priors are set to default), $T = 200$, $\lambda_u = 0.01$, $\lambda_v = 100$, $a = 1$, $b = 0.01$.

# 6 RESULTS AND DISCUSSION

## 6.1 Study I: Single Domain recommendations

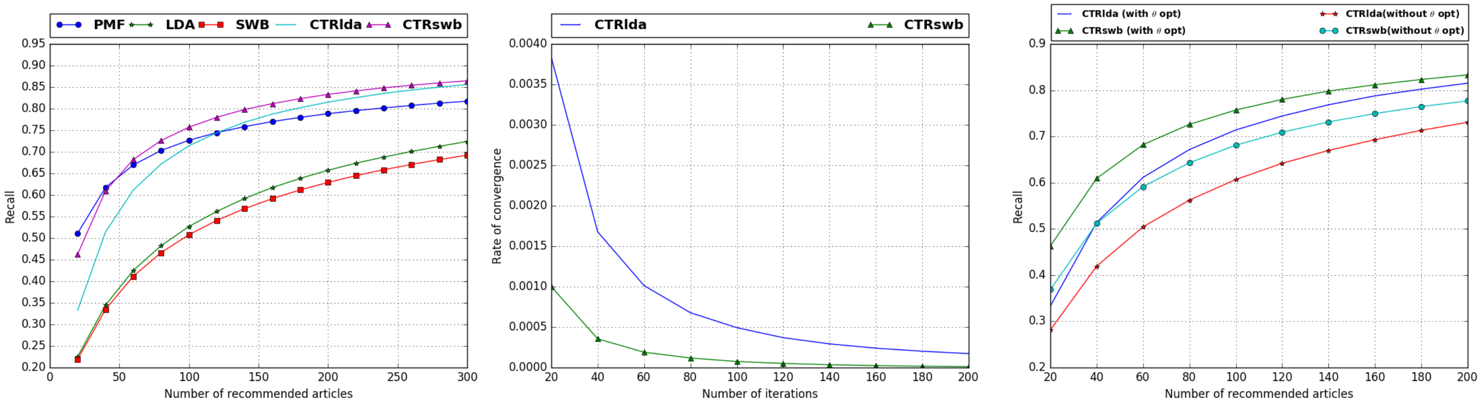
In this set of experiments, we compare the performance of the probabilistic matrix factorization (PMF), CTR model [22] which make use of latent topics from LDA (CTRlda), and the proposed CTRswb model. Fig. 3a shows our results on CiteULike dataset under the settings defined in [22]. In the graph, we also show how the topic proportion from LDA and SWB alone (i.e, when the user rating patterns from the train set are not considered) make predictions on the test set for *topK* (from 20 to 300) recommendations.

We can see that CTRswb consistently gives better recommendations than other factor models for different number of recommendations. Moreover, the margin of improvement for smaller number of recommendations is large between the CTRswb and CTRlda methods. Clearly, the PMF model lacks the content information and the pure content based models do not utilize user preferences and therefore, under-perform w.r.t CTR based models.

Further, we also show the performance of CTR based methods when subjected to iterative optimization of the parameter $\theta_j$. We observe that the CTRswb model has a faster convergence compared to CTRlda model as plotted in Fig. 3b. Clearly, the error gap analysis shows that the latent topics transferred from SWB model are in agreement with the consistent performance improvement of CTRswb methods over the CTRlda.

In Fig. 3c, we show the performance of CTR based methods both with and without $\theta_j$ optimization. The reason for CTRswb method giving the best performance, in both cases, is that in the real world item descriptions there will be lot of item specific terms, which will not be that much helpful for the recommendations. By removing the background terms of the corpus and specific terms from each items, we could aggregate the $\theta_j$ value in a precise manner.

[2] http://www.citeulike.org/
[3] http://www.imdb.com



**(a) Recall measure**

**(b) Convergence curve**

**(c) $\theta$ optimization**

**Figure 3: CiteULike Dataset in single domain recommendations: (a) Comparison for different recommendation algorithms in terms of recall measure. (b) Convergence curve of CTRlda and CTRswb w.r.t no. of iterations during $\theta$ optimization. (c) Performance of CTR based methods w.r.t $\theta$ optimization in terms of recall measure. (Best viewed in color)**

## 6.2 Study II: Cross Domain recommendations

In the cross-domain settings, we consider every genre in the dataset as a target domain while the other domains are treated as its auxiliary domains. For example, if "Action" genre is the target domain, the other four genres will constitute as the source domains.

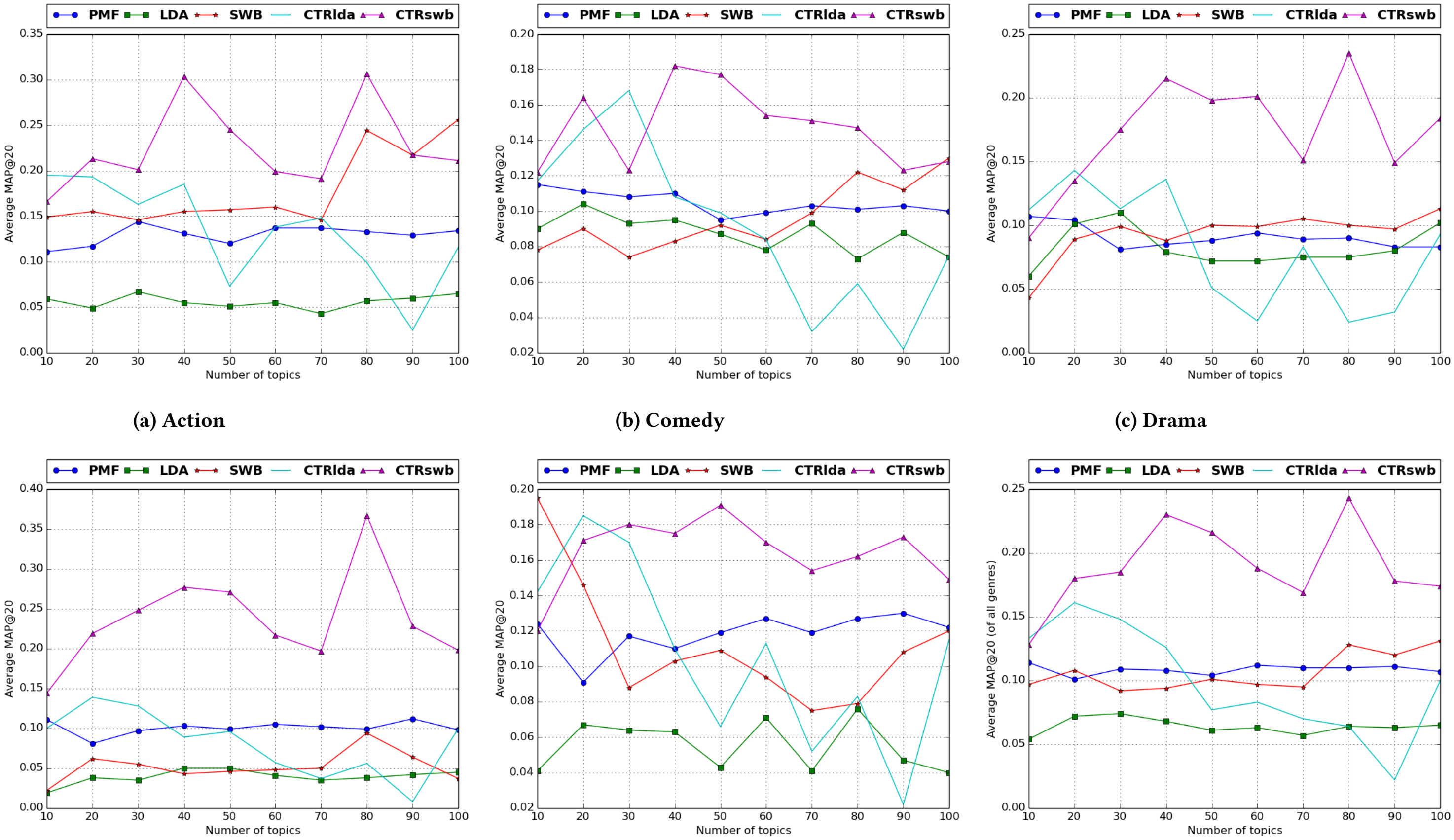
*6.2.1 Cold-start scenario and the impact of number of topics:* In this study, we consider the scenario when zero-rating information from the target domain while learning the latent topic features. From Table 2, we pick one of the genres as the target domain and create five cold-start scenarios (one for each genre in the dataset). We have run the algorithms PMF, LDA, SWB, CTRlda, CTRswb for each of the cold-start situations.

Figs. 4a–4e show mean average precision for *top*20 recommendations for five target genres. We can see that the MAP score of PMF model did not improve much when the number of latent factors are increased. Notice that, in many cases, the CTRlda method degrades the quality of recommendations when compared to traditional PMF. *Moreover, the CTRlda is highly sensitive to the number of latent factors and we noticed it consistently perform worse than the CTRswb. This could be reasoned as one of the potential problems with the learned topics that are obtained by feature fusion from multiple domains.* The CTRswb approach explicitly models these aspects and provides ability to improve the latent features. As we can see in the picture, our model consistently produces better quality of recommendations for different number of latent factors.

Fig. 4f shows the performance when averaged over all genres. From the plot, we observed that using 80 latent factors showed best performance for all genres except for comedy genre. The deviation in the case of "comedy" genre is expected as the number of items in the source domains are relatively less. Table 3 shows the performance of the different recommendations algorithms when 80 latent topics are used. Clearly, the proposed CTRswb model significantly improves over CTRlda and other methods in all the cold-start scenarios.

**Table 3: MovieLens Dataset: Comparison of different recommendation algorithms in terms of MAP, Precision (P), Recall (R). Here, we show the performance with 80 latent factors for all the five cold start scenarios. Bold numbers indicate best performance for a given target domain.**

| Genre | Method | MAP@20 | P@10 | P@20 | R@10 | R@20 |
|---|---|---|---|---|---|---|
| Action | PMF | 0.133 | 0.072 | 0.069 | 0.013 | 0.024 |
| | LDA | 0.057 | 0.025 | 0.026 | 0.005 | 0.01 |
| | SWB | 0.244 | 0.136 | 0.11 | 0.035 | 0.052 |
| | CTRlda | 0.099 | 0.061 | 0.057 | 0.013 | 0.025 |
| | CTRswb | **0.306** | **0.176** | **0.14** | **0.051** | **0.07** |
| Comedy | PMF | 0.101 | 0.05 | 0.049 | 0.008 | 0.014 |
| | LDA | 0.073 | 0.024 | 0.027 | 0.004 | 0.009 |
| | SWB | 0.122 | 0.059 | 0.05 | 0.009 | 0.016 |
| | CTRlda | 0.059 | 0.029 | 0.026 | 0.007 | 0.012 |
| | CTRswb | **0.147** | **0.074** | **0.061** | **0.011** | **0.018** |
| Drama | PMF | 0.09 | 0.039 | 0.038 | 0.006 | 0.012 |
| | LDA | 0.075 | 0.027 | 0.027 | 0.004 | 0.009 |
| | SWB | 0.1 | 0.044 | 0.041 | 0.009 | 0.016 |
| | CTRlda | 0.024 | 0.011 | 0.013 | 0.001 | 0.004 |
| | CTRswb | **0.235** | **0.07** | **0.055** | **0.02** | **0.026** |
| Romance | PMF | 0.099 | 0.048 | 0.046 | 0.015 | 0.028 |
| | LDA | 0.038 | 0.012 | 0.014 | 0.004 | 0.009 |
| | SWB | 0.094 | 0.029 | 0.025 | 0.024 | 0.037 |
| | CTRlda | 0.056 | 0.036 | 0.024 | 0.022 | 0.027 |
| | CTRswb | **0.367** | **0.084** | **0.06** | **0.061** | **0.07** |
| Thriller | PMF | 0.127 | 0.063 | 0.06 | 0.016 | 0.029 |
| | LDA | 0.076 | 0.035 | 0.028 | 0.012 | 0.018 |
| | SWB | 0.079 | 0.041 | 0.04 | 0.01 | 0.02 |
| | CTRlda | 0.084 | 0.038 | 0.031 | 0.016 | 0.027 |
| | CTRswb | **0.162** | **0.09** | **0.073** | **0.022** | **0.034** |



**(a) Action**

**(b) Comedy**

**(c) Drama**

**(d) Romance**

**(e) Thriller**

**(f) Mean of all Genres**

**Figure 4: Movielens Dataset in cross-domain recommendations: Impact of different sizes of latent topic space on quality of recommendations. Here, we use one of the genre as target domain, and remaining four as source domain. (Best viewed in color)**

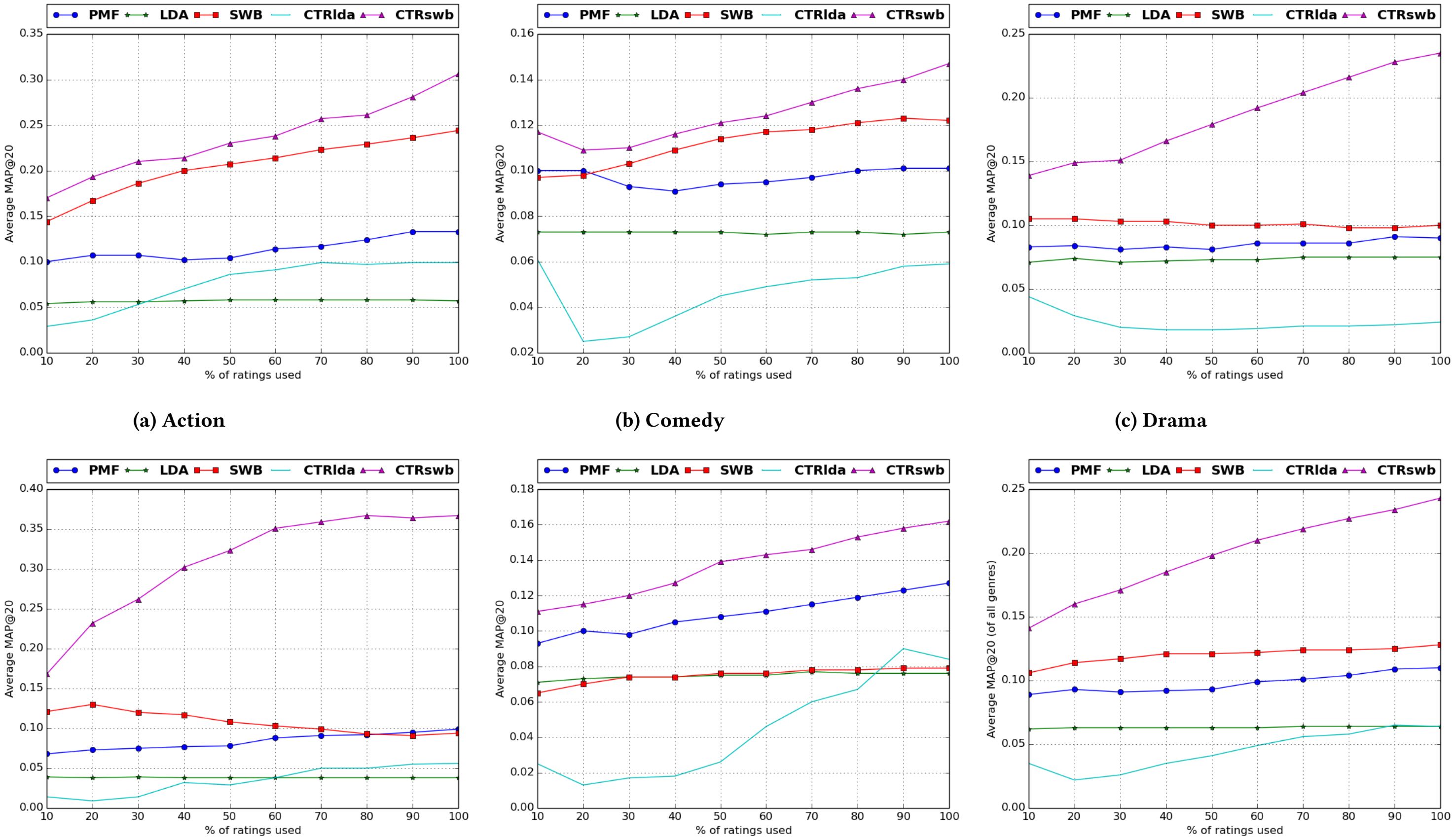
*6.2.2 Data sparsity scenario and the impact of number of ratings:* In this study, to explore the behavior of cross-domain recommendation, we examined the latent topic space under data sparsity scenario. We use the same movielens data as in Table 2 and create 10 data sparsity situations by incrementally removing (random) 10% of the ratings from the source genres. Throughout, similar to study in cold-start context, we do not use ratings from the target genre. To validate our findings, we have shown the evaluations only for the topic space of 80 latent factors. Figs. 5a–5e shows mean average precision of *top*20 recommendations for different degrees of sparsity (rating ratio) in the source domain.

The effect of number of ratings is much clear and straightforward compared to the effects of number of latent factors. The results reveal that the number of ratings in source genres have a significant impact on the accuracy. However, the scale of the impact is very different on each target domain as number of ratings in some genres are less. From the plots, it shows that the more user preferences are available in auxiliary domains, the better the accuracy of recommendations on target domain. When the number of ratings have increased, the PMF, LDA, SWB and CTRlda have shown moderate improvements in terms of MAP. Our approach consistently shows better performance, by large margin, than these methods. Over all, the results show that the latent factors of CTRswb are very reliable and could improve the recommendations even under extreme sparse data scenarios.

## 7 CONCLUSIONS

We have proposed an approach to validate our hypothesis that the quality of recommendations can be improved by explicitly utilizing the general topic word distributions while learning the latent features. Our approach recommends items to users based on both content and user preferences, and could at best exploit the content information in both single and cross-domain scenarios. Our results on single-domain show the superiority over pure latent factor and CTRlda models, and results on the cross-domain demonstrate its robustness under cold-start and data sparsity situations.

In the future, we plan to explore cross-domain recommendation scenarios in heterogeneous settings (e.g movies to books). In addition to this, we have used a simple collaborative filtering approach with zero-rating information from target domain, we believe utilizing the target domain ratings could result in better cross-domain recommendations.



**(a) Action**

**(b) Comedy**

**(c) Drama**

**(d) Romance**

**(e) Thriller**

**(f) Mean of all Genres**

**Figure 5: Movielens Dataset in cross-domain recommendations: Impact of the quality of recommendations for different amounts of ratings from the source domains. Here, we show the stability of CTRswb method when subjected to both cold-start (w.r.t target domain) and data sparsity (w.r.t source domains) scenario. (Best viewed in color)**

# REFERENCES

[1] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* (2003).

[2] Iván Cantador, Ignacio Fernández-Tobías, Shlomo Berkovsky, and Paolo Cremonesi. 2015. Cross-domain recommender systems. In *Recommender Systems Handbook*. Springer.

[3] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2007. Modeling general and specific aspects of documents with a probabilistic topic model. In *NIPS*.

[4] Wei Chen, Wynne Hsu, and Mong Li Lee. 2013. Making recommendations from multiple domains. In *ACM SIGKDD*.

[5] F Maxwell Harper and Joseph A Konstan. 2016. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* (2016).

[6] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *ICDM*.

[7] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *ACM REcSys*.

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

[9] Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *ICML*. ACM.

[10] Sujian Li, Jiwei Li, Tao Song, Wenjie Li, and Baobao Chang. 2013. A novel topic model for automatic term extraction. In *ACM SIGIR*.

[11] Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *ACM RecSys*.

[12] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *NIPS*.

[13] Orly Moreno, Bracha Shapira, Lior Rokach, and Guy Shani. 2012. Talmud: transfer learning for multiple domains. In *Proceedings of the 21st ACM international conference on Information and knowledge management*.

[14] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *ICDM*.

[15] Weike Pan, Nathan N Liu, Evan W Xiang, and Qiang Yang. 2011. Transfer learning to predict missing ratings via heterogeneous user feedbacks. In *IJCAI*.

[16] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. 2010. Transfer Learning in Collaborative Filtering for Sparsity Reduction. In *AAAI*.

[17] Steffen Rendle. 2010. Factorization machines. In *ICDM*.

[18] Shaghayegh Sahebi and Trevor Walker. 2014. Content-Based Cross-Domain Recommendations Using Segmented Models. *CBRecSys* (2014).

[19] Yue Shi, Martha Larson, and Alan Hanjalic. 2011. Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer.

[20] Mark Steyvers and Tom Griffiths. 2007. Probabilistic topic models. *Handbook of latent semantic analysis* (2007).

[21] Fatemeh Vahedian and Robin D Burke. Predicting Component Utilities for Linear-Weighted Hybrid Recommendation.

[22] Chong Wang and David M Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *ACM SIGKDD*.

[23] Tong Zhao, Julian McAuley, and Irwin King. 2015. Improving latent factor models via personalized feature projection for one class recommendation. In *ACM CIKM*.