Transferring User Interests Across Websites with Unstructured Text for Cold-Start Recommendation

Yu-Yang Huang and **Shou-De Lin**
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
{r02922050,sdlin}@csie.ntu.edu.tw

Abstract

In this work, we investigate the possibility of cross-website transfer learning for tackling the cold-start problem. To address the cold-start issues commonly present in a collaborative filtering (CF) system, most existing cross-domain CF models require auxiliary rating data from another domain; nevertheless, under the cross-website scenario, such data is often unobtainable. Therefore, we propose the nearest-neighbor transfer matrix factorization (NT-MF) model, where a topic model is applied to the unstructured user-generated content in the source domain, and the similarity between users in the latent topic space is utilized to guide the target-domain CF model. Specifically, the latent factors of the nearest-neighbors are regarded as a set of pseudo observations, which can be used to estimate the unknown parameters in the model. Improvement over previous methods, especially for the cold-start users, is demonstrated with experiments on a real-world cross-website dataset.

1 Introduction

While collaborative filtering (CF) approaches are one of the most successful methods for building recommender systems, their performance deteriorates dramatically under *cold-start* situations. That is, low prediction accuracy is observed for users/items with very few ratings. Content-based recommender systems may also suffer from the cold-start problem. For instance, content-based nearest-neighbor models (Pazzani and Billsus, 2007) might not be as effective if some users contain too few information to generate a meaningful set of neighbors.

Two types of solutions have been proposed to address the cold-start problem. The first is to create hybrid recommendation models that impose a content-based model on a CF model to enrich the information for users/items with sparse rating profiles (Burke, 2002; Burke, 2007). The second is to transfer the information from auxiliary domains as a remedy to the cold-start individuals (Deng et al., 2015). This paper aims at bringing a marriage between these two types of strategies.

Although transfer learning gradually gains popularity in handling the cold-start issue (Roy et al., 2012), most of them assume a homogeneous model where observations in both domains are of the same type. That is, to transfer knowledge to a rating-based/text-based recommender system, the source system must also be rating-based/text-based. Some earlier works even require the ratings from both domains to be in the same format (Li et al., 2009), or assume specific structured text, such as user-provided tags (Shi et al., 2011; Deng et al., 2015). In this work, by contrast, no source-domain ratings are available and unstructured user-generated content is treated as the auxiliary data. We propose a *heterogeneous transfer learning* framework to utilize unstructured auxiliary text for a better target-domain CF model.

As there is no single service satisfying all social needs, users nowadays hold multiple accounts across many websites. Furthermore, the account linking mechanism is often available on these websites. This allows a precise mapping between the accounts of the same user to be built. One major application of our approach is to improve the recom-

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 805–814,
Austin, Texas, November 1-5, 2016.

mendation quality in the target service using auxiliary data obtained from another seemingly irrelevant service.

For instance, consider a new user on YouTube. The initial recommended videos for this user is likely to be irrelevant as there is very few information available. However, with the account linking mechanism, YouTube accounts can be linked to Twitter accounts with a simple click. Our goal is to utilize the content generated by this user on Twitter, despite the possibility that the content is irrelevant to their preference on video browsing, to produce a better video recommendation list on YouTube.

Seemingly intuitive, there exist some difficulties in such *cross-website transfer learning* approach. The biggest challenge lies in the fact that most users do not use multiple services (e.g. social media sites) for the same purpose. Usually a user registers for multiple services because each of them serves its own purpose. As a result, we cannot assume the existence of direct mentions about target-domain items in the source-domain text data. For example, a regular YouTube viewer does not necessarily tweet about the videos he/she has viewed. Thus simple methods such as keyword matching are likely to fail. The same reasoning also implies that, when transferring knowledge across websites or services, the assumption of a shared rating format or structured text is overly optimistic. Even websites aiming for the same purpose often violate this assumption, let alone websites of different types. Therefore, we expect that the source and target services contain heterogeneous information (e.g. content vs. rating). In our model, we make a less strong assumption: regardless of the type of information available in each domain, the users that are similar in one domain should have similar taste in the other domain. Thus, instead of directly transfer the content material from source to target domain, we transfer the *similarity* between users, and use it as a guide to improve the CF model in the target domain.

2 LDA-MF Model

We first introduce an intuitive model to realize the above-mentioned idea, and point out several intrinsic weaknesses making it unsuitable for cross-website transfer learning.

Here we rely on the probabilistic matrix factorization (PMF) model as our target-domain CF model. In the PMF model, each user latent factor is modeled (a priori) by a zero-mean Gaussian. To incorporate source-domain information into the target-domain PMF model, for each user i, a topic vector θ_i is extracted from source-domain text content and assigned as the prior mean of this user's PMF latent factor, that is,

$$u_i \sim \mathcal{N}(\theta_i, \lambda_U^{-1} I), \tag{1}$$

where λ_U is the precision parameter and I is the identity matrix. Different from the original PMF model, prior distributions of different user latent factors are no longer identical. For users having similar source-domain topic vectors, their latent factors are expected to be close in the target-domain latent space. Such property allows the similarity between users to be transferred from source domain to the target domain.

With the latent Dirichlet allocation (LDA) (Blei et al., 2003) topic model being used, the graphical model is depicted in Figure 1. This model is similar in structure to the recently proposed collaborative topic regression (CTR) (Wang and Blei, 2011) model. The main difference is that, instead of modeling description about items, now user-generated content *from the source domain* is modeled in our problem. We call this model the LDA-MF model.

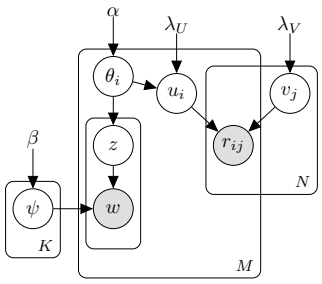


Figure 1: The LDA-MF model.

Although LDA-MF indeed incorporates knowledge from the source domain, it has certain weaknesses which need to be addressed. The most significant drawback is that the dimensionalities of the LDA topic vector θ_i and the PMF user latent factor u_i are required to be equal. These two variables are of very different nature. One is extracted from text data in the source domain to model the topics

of the user-generated content, and the other is generated from the rating data in the target domain to model the latent interests of users. It is an overly strong assumption to assume the optimal dimensionalities for LDA and PMF are equal. In practice, if we choose the dimensionality to optimize the predictive power of PMF (e.g. by cross-validation on the rating data), the LDA model is likely to yield sub-optimal results and vice versa. The experiments that will be shown later confirm this concern. Furthermore, since the two variables are modeling different types of observations coming from different websites, the underlying meanings of the latent dimensions are unlikely to be identical. By treating the LDA topic vector as the prior mean of the PMF user latent factor, the latent dimensions are forced to be one-to-one aligned, which is again a strong assumption. Finally, the topic vectors are drawn from the Dirichlet distribution which has a bounded (and positive) support S, while the latent factors in PMF are unbounded Gaussian random vectors. If the optimal solution of u_i is far from S, the performance of the model could be affected, particularly in the cold-start situation where data is sparse and the prior plays an important role.

3 Nearest-Neighbor Transfer MF Model

To alleviate the drawbacks of the LDA-MF model, here we propose *nearest-neighbor transfer matrix factorization* (NT-MF) model to transfer user interests across websites. The entire framework is depicted in Figure 2.

We begin by describing the scenario in which our model operates. First, there is a rating-based recommender system (i.e. PMF) in the target domain, which suffers from the cold-start problem. The target domain might or might not contain content information. For example, in the video recommendation task, we can use the titles of all rated videos of a user to generate content information in the target domain. Such information is not available for the cold-start users since they have not rated any videos. However, in the source domain there are some content information available for these users. This can be, say, the content of a user's tweets. As previously mentioned, this type of auxiliary text data is immediately available when a user connects the accounts

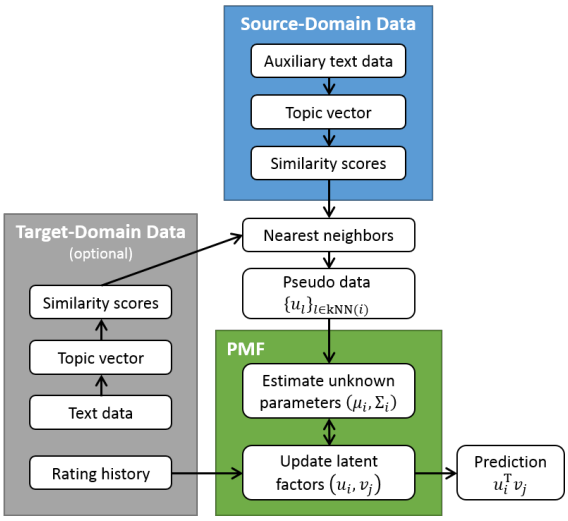


Figure 2: The entire system.

from two domains. Therefore, we assume this auxiliary text data is available for all users.

3.1 Model Outline

Next, we describe the high level concept of our model. As described previously, we have observed that the hypotheses encoded by the LDA-MF model is too strong as the PMF latent factor is enforced to inherit certain mathematical properties from the LDA topic vector. Here we loosen the constraint to only enforce that users should have similar distributions over the target-domain PMF latent factors if there is a high similarity between their source-domain topic vectors.

It is a reasonable hypothesis since our objective is to make the target-domain rating matrix factorize in a way that is consistent with the knowledge extracted from source-domain text. After all, the factorization of incomplete matrix is not unique, and there is no reason that the latent factor should match the topic factor of the user. In fact, our hypothesis implies a different distribution over the PMF latent factor for each user, i.e. $u_i \sim \mathcal{N}(\mu_i, \Sigma_i)$, where (μ_i, Σ_i) are unknown parameters, and are (possibly) different for each user.

To estimate the unknown parameters in a distribution, normally we need a set of observations, $(u_i^{(1)}, u_i^{(2)}, \dots)$. However, the parameters now belong to a distribution over a latent variable, which is non-trivial to estimate since we have no observations about the user latent factor. An exhaustive search

over the parameter space is obviously intractable. Even if we treat the entire model as a hierarchical model and learn the parameters indirectly from rating data, the cold-start problem immediately comes in and forbids us from learning a representative distribution for users.

We propose the idea of using the latent factors of the nearest-neighbors to estimate the unknown parameters in the distribution for a user. That is, the latent factors of the nearest-neighbors, $\{u_l\}_{l \in k\text{NN}(i)}$, are regarded as a set of *pseudo observations* to replace the unavailable data, $(u_i^{(1)}, u_i^{(2)}, \dots)$. However, the definition of "closeness" is not based on target-domain rating data, but computed by the topic vectors obtained from the content in the source domain (and the target domain, if available). Our model is thus not hampered by the cold-start problem.

Note that, in addition to the set of k-nearest-neighbors $k\text{NN}(i)$, we also have the corresponding similarity scores $sim(i, l)$ between each neighbor l and user i. The similarity scores along with the list of nearest-neighbors are transferred to the target domain to form a set of weighted samples, $\mathcal{D}$, which can be used to estimate the unknown parameters (μ_i, Σ_i), i.e.,

$$\mathcal{D} \equiv \begin{cases} \{u_l\}_{l \in k\text{NN}(i)} \\ w_l = sim(i, l). \end{cases} \tag{2}$$

The main purpose of assigning a sample weight w_l to each of the pseudo observations u_l is that by doing so, users with a higher source-domain similarity to user i will have a larger impact on the estimation of the target-domain parameters (μ_i, Σ_i). In other words, with this model specification, the *similarity between users* is transferred across domains.

3.2 Inference in NT-MF Model

To perform inference in our model, we adopt the maximum a posteriori (MAP) strategy and alternately update the user and item latent factors (i.e. by block coordinate ascent), similar to some previous solutions (Salakhutdinov and Mnih, 2007; Wang and Blei, 2011).

To solve for the optimal user latent factor u_i, we need to first estimate the unknown parameters (μ_i, Σ_i). Therefore, in our coordinate ascent algorithm, different from the original PMF model, we update the user latent factors one by one. That is, all user latent factors are regarded as fixed constants except for the one, u_i, to be updated. By doing so, for each user i, a set of *pseudo observations* about u_i (Eq. 2) is available. Using these pseudo observations, the unknown parameters (μ_i, Σ_i) can then be estimated with standard techniques such as maximum likelihood estimation (MLE). After an estimator of (μ_i, Σ_i) is obtained, we can analytically solve for the MAP solution of the user latent factor u_i. Then, we move on to the next user, and the coordinate ascent procedure continues. These two steps, namely the estimation of unknown parameters and the updating of the latent factors, are repeated until convergence.

One advantage of this procedure is that the list of nearest-neighbors and the similarities in Eq. 2 need not be recomputed during inference, avoiding expensive recomputation of pairwise similarities. It is also noticeable that, different from other transfer-based approaches, rating information and structured text from the source domain are not required in this procedure of model optimization. This further adds a level of flexibility to our framework for transferring user interests across websites.

3.3 Case Study: Inferring Unknown Mean

To clarify the previous discussions, we present a simple but detailed case-study on how an NT-MF model and its optimization procedure can be derived. The latent factor u_i for each user is assumed to be generated from a multivariate normal distribution with unknown mean μ_i and a known precision parameter λ_U, which is shared among the users.

The generative process proceeds as follows:

1. For each user i, draw user latent factor $u_i \sim \mathcal{N}(\mu_i, \lambda_U^{-1} I)$.
2. For each item j, draw item latent factor $v_j \sim \mathcal{N}(0, \lambda_V^{-1} I)$.
3. For each observed user-to-item pair (i, j), draw the rating $r_{ij} \sim \mathcal{N}(u_i^T v_j, \lambda_0^{-1})$,

where λ_0 is the precision parameter of the ratings, and λ_U, λ_V are the precision parameter of the

users and items, respectively. We use the notation $\mathcal{N}(x|\mu, \Sigma)$ to denote the Gaussian pdf with mean μ and covariance Σ.

The model is optimized by maximizing the posterior likelihood of the latent variables (an additive term is omitted),

$$\mathcal{L} = -\frac{\lambda_0}{2}\sum_{i=1}^{M}\sum_{j=1}^{N}\gamma_{ij}\left(r_{ij} - u_i^T v_j\right)^2 - \frac{\lambda_U}{2}\sum_{i=1}^{M}\left(u_i - \mu_i\right)^T\left(u_i - \mu_i\right) - \frac{\lambda_V}{2}\sum_{j=1}^{N} v_j^T v_j, \tag{3}$$

where γ_{ij} is an indicator variable which is equal to 1 if item j is rated by user i, and 0 otherwise.

To solve the MAP problem, we need to first estimate the unknown parameters in the distribution, which in this case is the mean vector μ_i. The likelihood function over the pseudo observations, $\{u_l\}_{l\in k\text{NN}(i)}$, is defined as,

$$p(\mathcal{D}|\mu_i, \lambda_U) = \prod_{l\in k\text{NN}(i)} \mathcal{N}(u_l|\mu_i, \lambda_U^{-1} I). \tag{4}$$

By taking derivative of Eq. 4 with respect to μ_i and set it to zero, we obtain,

$$\sum_{l\in k\text{NN}(i)} (u_l - \mu_i) = 0, \tag{5}$$

which implies that the MLE of μ_i is the sample mean. However, since we are dealing with a set of *weighted samples*, the sample mean is replaced by the weighted average (the weights w_l are assumed to add up to one):

$$\mu_i = \sum_{l\in k\text{NN}(i)} w_l u_l. \tag{6}$$

Our model yields an intuitive result: to estimate the mean vector μ_i of u_i, we can simply take the weighted average of the latent factors u_l from the nearest-neighbors as an estimator, where the weights are the similarity scores between the textual profiles of user i and its neighbors.

Given μ_i, we can now maximize Eq. 3 with respect to u_i and v_j. By taking derivative of Eq. 3 with respect to u_i and v_j and set it to zero, we obtain the update equations,

$$\left(\sum_{j=1}^{N}\gamma_{ij} v_j v_j^T + \frac{\lambda_U}{\lambda_0} I\right) u_i = \sum_{j=1}^{N}\gamma_{ij} r_{ij} v_j + \frac{\lambda_U}{\lambda_0}\mu_i \tag{7}$$

$$\left(\sum_{i=1}^{M}\gamma_{ij} u_i u_i^T + \frac{\lambda_V}{\lambda_0} I\right) v_j = \sum_{i=1}^{M}\gamma_{ij} r_{ij} u_i. \tag{8}$$

Now with Eq. 6 to Eq. 8 at hand, we can iteratively solve for μ_i, u_i and v_j for all users and items until the model converges.

It can be seen from this case-study that NT-MF eliminates the three major drawbacks of the previously mentioned LDA-MF model. First, the topic vectors and the user latent factors are not required to have equal dimensionalities, which allows for the optimal dimensionality to be chosen in both models. Second, the mean vector, that is, the kNN weighted average in Eq. 6, is a linear combination of a set of user latent factors; as a result, the latent dimensions of u_i and μ_i are naturally aligned. Third, the mean vector μ_i has the same support as the user latent factor u_i, avoiding the risk of prior misspecification in cold-start situations.

4 Experiment

We use YouTube video recommendation to test the usefulness of NT-MF under the cold-start scenario. The NT-MF model used in this section follows the optimization procedure derived in Section 3.3.

4.1 Dataset and Statistics

To construct a dataset containing both the users' rating history and textual information, we begin with the user profile pages on Google+. A large proportion of Google+ users provide links to their profile pages from other social network services (e.g. Twitter). More importantly, if a user owns a YouTube account, a link to the user's YouTube channel will be automatically added to his Google+ profile. This makes a fully aligned dataset available. Users' Twitter accounts are obtained via their Google+ profile page, and the concatenation of tweets is regarded as the auxiliary text data. It has been shown that by concatenating the tweets, more representative user

topic vectors can be obtained (Hong and Davison, 2010). We refer to this text data as the *Twitter corpus*.

Videos in a user's "liked" or "favorite" playlists are considered to have a rating $r_{ij} = 1$. Other videos are assigned $r_{ij} = 0$. In other words, we are dealing with a one-class collaborative filtering (OCCF) problem (Pan et al., 2008). We adopt the same strategy as in (Wang and Blei, 2011) to deal with OCCF. First, all ratings are assumed to be observed, i.e. $\gamma_{ij} = 1$ for all user-item pairs. Next, a confidence parameter c_{ij} is introduced to reduce the influence of the huge number of zeroes during model optimization. The confidence parameter takes place of the original rating precision parameter λ_0 and is defined in (Wang and Blei, 2011) as $c_{ij} = a$ if $r_{ij} = 1$ and $c_{ij} = b$ otherwise $(a > b > 0)$. All the derivations in the previous sections follow intuitively.

The titles of the liked videos are concatenated and treated as the text data in the target domain (which we refer to as the *YouTube corpus*). As for the vocabulary, stopwords are first removed, and then 5000 words are selected from the YouTube corpus based on their TF-IDF scores (Blei and Lafferty, 2009). On average, each user's Twitter text data contains 5149 words and 1193 distinct terms, and each user's YouTube text data contains 158 words and 116 distinct terms. These statistics are in accordance with our assumption that text data in the source domain is abundant comparing to that in the target domain.

To validate the prediction result, each user has at least 10 liked videos. Videos with less than 5 likes are removed from the dataset. After data cleansing, there are 7328 users and 18691 videos in the dataset. The maximum number of likes received by a video is 98, and the average is 19.1. Among all videos, 92% of them are liked by less than 40 users. The maximum number of likes given by a user is 908, and the average is 48.8. Among all users, 89% of them have liked less than 100 videos. The sparsity (ratio of zeroes to the total number of entries) of the rating matrix is 99.74%, which illustrates the difficulty of this recommendation task.

4.2 Evaluation and Scenario

We choose the area under ROC curve (AUC) as the evaluation metric. AUC is often used to compare models when there is severe class imbalance, which is the case in our OCCF problem since we regard all zeroes as observed. All reported results are the average of 5 random data splits.

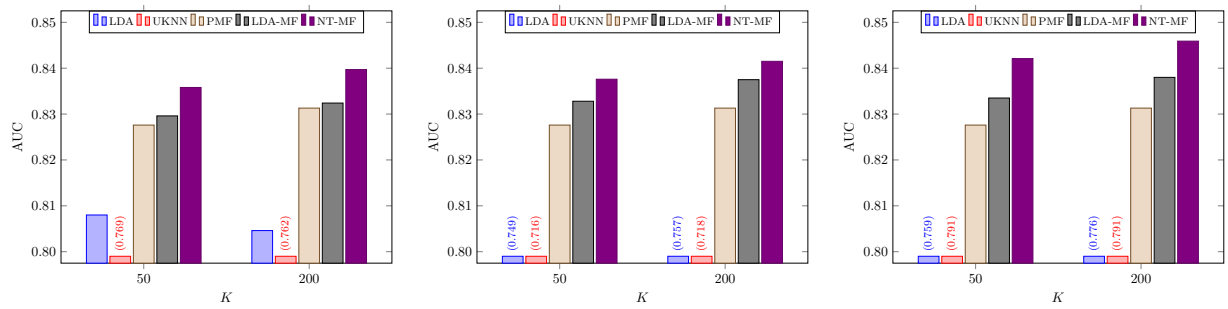
Similar to the experiments performed in (Wang and Blei, 2011), we test the performance of each model under two different scenarios. The first one is the task of *in-matrix* prediction. In this task, the likes received by each video are partitioned into three sets, namely the training, validation and testing sets. The ratio of data partition is 3:1:1. There are no cold-start users for the in-matrix prediction.

The second task is the *out-of-matrix* prediction, where the *users* are partitioned into three sets with the same 3:1:1 ratio. To make the two tasks comparable, we randomly split the data until the number of observations in each of the three sets is closed to that of the in-matrix task. Users in the testing set are all cold-start users. The only data we have when making prediction on the cold-start users is the auxiliary text data.

4.3 Baseline Methods

- **LDA:** We run linear regression on the LDA features to predict the ratings. This model serves as a content-based baseline.
- **UKNN:** The user-kNN algorithm (Herlocker et al., 1999) based on LDA features is implemented. This model serves as a neighborhood-based baseline.
- **PMF:** PMF (Salakhutdinov and Mnih, 2007) is a classic and widely-used CF model. It uses only the rating information, and thus is not capable of performing the out-of-matrix task.
- **LDA-MF:** This model is implemented as has been described in Section 2. It is similar to CTR (Wang and Blei, 2011) in structure. Since the optimization of the full model converges badly, we pre-train the LDA part of the model, and fix the topic vector when optimizing the PMF part.

All hyperparameters are tuned on the validation set. Due to efficiency and storage considerations, for UKNN and NT-MF, the k-nearest-neighbors are computed approximately with the FLANN library



(a) YouTube corpus (b) Twitter corpus (c) YouTube + Twitter corpus

Figure 3: In-matrix AUC using different corpus. For methods significantly worse than others, we cut off the plot and put the AUC values on top of the bars. NT-MF is significantly better than the baselines in all plots, according to a paired t-test ($p < 0.05$).

(Muja and Lowe, 2014). The symmetric Kullback-Leibler divergence is chosen to be the distance metric between topic vectors. For all baseline methods, we use K to denote the dimensionality of the latent variables. However, when discussing about NT-MF, since the number of topics can be different from the number of user latent factors, we use T to denote the former and K to denote the latter to avoid confusion.

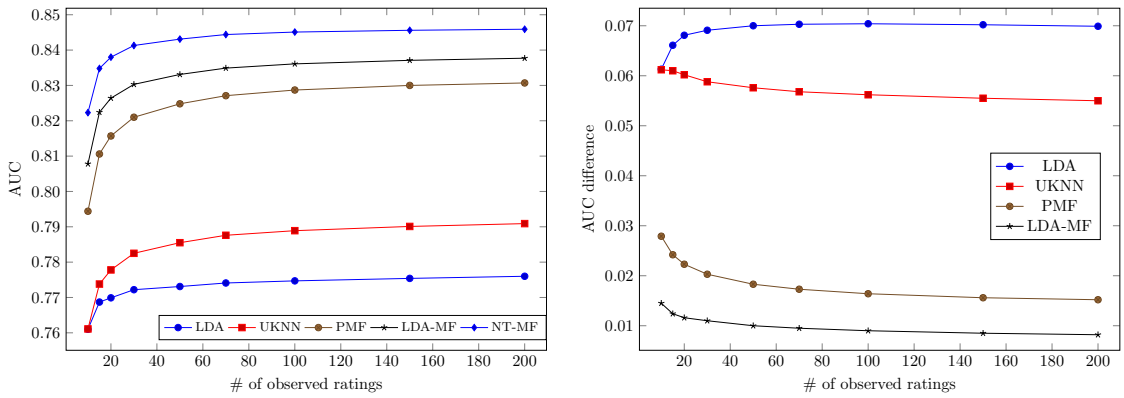
4.4 In-Matrix Prediction

In this section, the in-matrix prediction is discussed. First, we test the model's general performance on different corpora. Normally, the optimal number of topics will not be the same for different corpora. Since the LDA model performs the best with $K = 50$ on the YouTube corpus and $K = 200$ on the Twitter corpus, we report the results when K is set to these two numbers.

Figure 3(a) shows the results when no source-domain information is available and thus no transfer learning is performed. That is, all models are provided only with the YouTube ratings and the YouTube corpus. Because the YouTube corpus is scarce, the LDA model results in lower AUC when more topics are used, signifying overfitting. The same reason also leads to limited improvement of LDA-MF over PMF. Using neighborhood information alone, UKNN performs poorly. On the other hand, as a model bringing neighborhood information into PMF, NT-MF outperforms all baselines significantly. The above analysis shows that, although using either content (LDA) or neighborhood (UKNN) information alone is insufficient to generate good predictions, they can effectively improve the factorization of the rating matrix if used correctly.

To demonstrate the advantage of transfer learning, we study the scenario where only source-domain text and target-domain ratings are available. That is, the YouTube corpus in the previous analysis is replaced with the Twitter corpus. The result is shown in Figure 3(b). Comparing to Figure 3(a), we can see that although the Twitter corpus is larger than the YouTube corpus, it leads to a worse performance for LDA and UKNN. Content information from the noisy Twitter corpus alone is not sufficient to capture the rating behavior of users. However, by integrating the content information and rating history, both LDA-MF and NT-MF benefit from a larger corpus.

In the following analyses, we use data from both websites. For LDA, PMF and LDA-MF, we merge the two corpora by summing up the word counts. For UKNN and NT-MF, however, there is a more elegant way to combine the knowledge from different websites. First, we compute user similarity separately from the two corpora. Then, the two sets of similarity scores are weighted and averaged. Finally, the nearest-neighbors are computed based on this set of newly generated similarity scores. By applying this strategy to NT-MF, not only can θ_i and u_i differ in dimensionality, but also the optimal number of topics can be used for different corpora. Regardless of K, we use $T = 50$ for YouTube and $T = 200$ for Twitter in our NT-MF model. The result is shown in Figure 3(c). By comparing it with Figure 3(b), we can see that the AUC of NT-MF increases while that of LDA-MF remains unchanged. UKNN also benefits from this strategy. These facts show that, instead of merging the two corpora directly, our strategy of averaging the similarities is more advantageous.



(a) (b)

Figure 4: (a) Cumulative in-matrix AUC. Each point (x, y) in the figure means that the model gives an averaged AUC of y among all users who have less than or equal to x observed ratings. (b) Difference in cumulative in-matrix AUC between NT-MF and baseline methods.

Next, as a preliminary investigation of the performance on cold-start users, in Figure 4(a), we plot the *cumulative* AUC with respect to the total number of observed ratings. NT-MF outperforms other methods in terms of cumulative AUC regardless of the number of observed ratings. The advantage of NT-MF over the baseline methods is even greater as the number of observed ratings decreases (except for LDA). To make it clear, we plot the difference in AUC between NT-MF and the baseline methods in Figure 4(b). This phenomenon sheds light on the advantage of NT-MF under cold-start scenario.

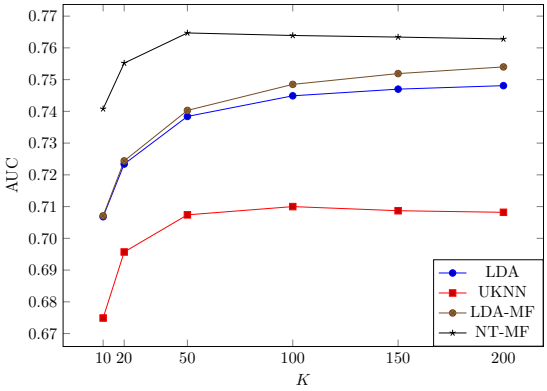


Figure 5: Out-of-matrix AUC. NT-MF is significantly better than the baselines, according to a paired t-test ($p < 0.05$).

4.5 Out-of-Matrix Prediction

In this section, we discuss the *out-of-matrix* prediction. Users in the testing set are all completely cold-start users. That is, we are only provided the Twitter corpus when making prediction for these users. Therefore, our previous strategy of averaging the similarities only applies to users in the training set. For this study we adopt the strategy of merging the two corpus instead of averaging the similarities. The number of topics $T = 150$ is chosen for NT-MF with respect to the validation AUC.

The result is presented in Figure 5. We plot the AUC against the dimensionality of the latent variables K. It can be observed that NT-MF beats all baseline methods regardless of K. Comparing to Figure 3, the out-of-matrix AUC is much lower, signifying the difficulty of cold-start recommendation.

Under the cold-start scenario, the latent factor used in the prediction phase is taken to be the prior mean for the MF-based models. For LDA-MF the prior mean is the topic vector θ_i, while for NT-MF it is the weighted average μ_i given by Eq. 6.

Since θ_i is used in place of u_i in the LDA-MF model when generating predictions, the curves of LDA and LDA-MF look very similar. A paired t-test ($p < 0.05$) shows no statistically significant difference between these two methods when $K = 10$ ($p = 0.48$) and $K = 20$ ($p = 0.09$). Despite the fact that $u_i = \theta_i$ is fixed for the cold-start users in the LDA-MF model, as K becomes larger, the item latent factors can carry more information in the rating data, which results in a higher AUC than LDA. However, since the dimensionalities of the LDA part and PMF part must match, the inference procedure of LDA-MF becomes very slow when K is large. To

make a better use of the available data, the computational efficiency must be sacrificed.

On the other hand, note that NT-MF achieves the highest AUC when $K = 50$. In fact, not only does NT-MF beat all baseline methods under different K values, it also outperforms the best LDA-MF model ($K = 200$) with fewer latent factors ($K = 20$). Unlike LDA-MF, the latent factors of the cold-start users are not fixed in NT-MF. Therefore, NT-MF can represent the information in a more concise way. In this case, NT-MF is better than LDA-MF in terms of both execution speed and predictive power.

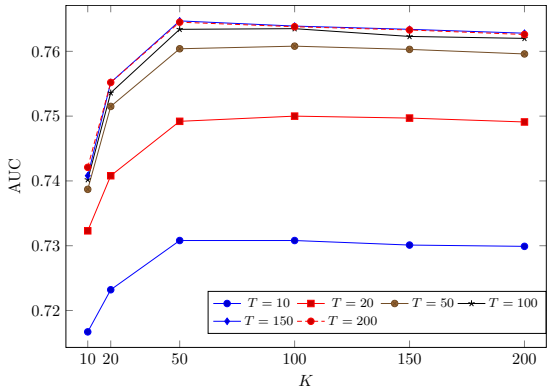


Figure 6: Performance of NT-MF based on out-of-matrix AUC for different values of K and T.

In Figure 6 we investigate the effect of different values of K and T. For each curve, we can see that the performance is about the same for $K \geq 50$. This is in accordance with the observation that NT-MF does not need as many latent factors as LDA-MF to achieve the same level of performance. Also, while increasing the number of topics T improves the performance in general, increasing T from 150 to 200 gives no significant improvement. The most important observation is that the highest AUC is achieved when $K = 50$ and $T = 150$. In other words, the optimal number of topics is different from that of user latent factors. This further justifies the advantage of NT-MF against previous methods.

5 Related Work

Although not directly aiming to solve the problem we have proposed, there exists some models of similar structure or adopt similar ideas.

As previously mentioned, LDA-MF is similar in structure to CTR. Collaborative topic Poisson factorization (CTPF) (Gopalan et al., 2014) combines the ideas of CTR and Poisson factorization (Gopalan et al., 2013) for a better performance. We have also tried CTPF on our dataset; nevertheless, there is no significant improvement over LDA-MF.

Recently, the neighborhood-aware probabilistic matrix factorization (NHPMF) model is proposed (Wu et al., 2012) as a method to combine kNN and PMF. It is originally proposed to leverage tagging data for improving PMF. This model can also be applied to our problem if we use the Twitter corpus in place of the unavailable tagging data. However, in the NHPMF model, the mean parameters are not treated as constants when the user latent factors are updated. As a result, an extra term appears in the gradient formula, which leads to an $O(k^2)$ time complexity, with k being the number of nearest-neighbors considered. On the other hand, the computation of the weighted average (i.e. Eq. 6) takes $O(k)$ time complexity. We have implemented NHPMF for comparison. As we increase k, NHPMF becomes significantly slower than NT-MF, while its performance is no better than NT-MF on our dataset.

6 Conclusion

In this work, we propose NT-MF, a cross-website transfer learning model which integrates content, neighborhood and rating information to alleviate the cold-start problem. A significant improvement over previous methods is demonstrated on a real-world cross-website dataset. The improvement is even more significant under the cold-start scenario.

So far we use the LDA topic vector to represent a user. As future work, different aspects of text can be taken into account to generate a more comprehensive user model. For example, writing styles or opinion mining may provide different insights on user behavior. Another possible extension is to apply our idea to more realistic settings such as large-scale and online recommender systems.

Acknowledgments

This material is based upon work supported by Microsoft Research Asia (MSRA) under award number FY16-RES-THEME-013 and by Taiwan Ministry of Science and Technology (MOST) under grant number 103-2221-E-002-104-MY2.

References

David M. Blei and John D. Lafferty. 2009. Topic models. In *Text Mining: Theory and Applications*. Taylor and Francis.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March.

Robin Burke. 2002. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, November.

Robin Burke. 2007. The adaptive web. chapter Hybrid Web Recommender Systems, pages 377–408. Springer-Verlag, Berlin, Heidelberg.

Zhengyu Deng, Ming Yan, Jitao Sang, and Changsheng Xu. 2015. Twitter is faster: Personalized time-aware video recommendation from twitter to youtube. *ACM Trans. Multimedia Comput. Commun. Appl.*, 11(2):31:1–31:23, January.

Prem Gopalan, Jake M. Hofman, and David M. Blei. 2013. Scalable recommendation with poisson factorization. *CoRR*, abs/1311.1704.

Prem Gopalan, Laurent Charlin, and David M. Blei. 2014. Content-based recommendations with poisson factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3176–3184. Curran Associates, Inc.

Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. 1999. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '99, pages 230–237, New York, NY, USA. ACM.

Liangjie Hong and Brian D. Davison. 2010. Empirical study of topic modeling in twitter. In *Proceedings of the First Workshop on Social Media Analytics*, SOMA '10, pages 80–88, New York, NY, USA. ACM.

Bin Li, Qiang Yang, and Xiangyang Xue. 2009. Can movies and books collaborate?: Cross-domain collaborative filtering for sparsity reduction. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, IJCAI'09, pages 2052–2057, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Marius Muja and David G. Lowe. 2014. Scalable nearest neighbor algorithms for high dimensional data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36.

Rong Pan, Yunhong Zhou, Bin Cao, Nathan Nan Liu, Rajan M. Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 502–511.

Michael J. Pazzani and Daniel Billsus. 2007. The adaptive web. chapter Content-based Recommendation Systems, pages 325–341. Springer-Verlag, Berlin, Heidelberg.

Suman Deb Roy, Tao Mei, Wenjun Zeng, and Shipeng Li. 2012. Socialtransfer: Cross-domain transfer learning from social streams for media applications. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 649–658, New York, NY, USA. ACM.

Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 1257–1264.

Yue Shi, Martha Larson, and Alan Hanjalic. 2011. Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering. In *Proceedings of the 19th International Conference on User Modeling, Adaption, and Personalization*, UMAP'11, pages 305–316, Berlin, Heidelberg. Springer-Verlag.

Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 448–456, New York, NY, USA. ACM.

Le Wu, Enhong Chen, Qi Liu, Linli Xu, Tengfei Bao, and Lei Zhang. 2012. Leveraging tagging for neighborhood-aware probabilistic matrix factorization. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1854–1858, New York, NY, USA. ACM.