Non-Linear Cross-Domain Collaborative Filtering via Hyper-Structure Transfer

Yan-Fu Liu YFLIU@NETDB.CS.NTHU.EDU.TW
National Tsing Hua University, ROC

Cheng-Yu Hsu CYHSU@NETDB.CS.NTHU.EDU.TW
National Tsing Hua University, ROC

Shan-Hung Wu SHWU@CS.NTHU.EDU.TW
National Tsing Hua University, ROC

Abstract

The *Cross Domain Collaborative Filtering* (CDCF) exploits the rating matrices from multiple domains to make better recommendations. Existing CDCF methods adopt the *sub-structure sharing* technique that can only transfer *linearly correlated* knowledge between domains. In this paper, we propose the notion of *Hyper-Structure Transfer* (HST) that requires the rating matrices to be explained by the *projections* of some more complex structure, called the *hyper-structure*, shared by all domains, and thus allows the *non-linearly correlated* knowledge between domains to be identified and transferred. Extensive experiments are conducted and the results demonstrate the effectiveness of our HST models empirically.

1. Introduction

Collaborative Filtering (CF) (Hofmann, 2004; Hu et al., 2008; Koren et al., 2009) is a major technique used by the recommender systems to find out items interesting to a user. The problem of CF can be defined as follows: given a rating matrix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ whose each entry $\boldsymbol{X}_{i,j}$ denotes either the rating (or other similar feedback) from the user i to the item j if i has interacted with j or blank otherwise, fill the blanks in $\boldsymbol{X}$ so that a recommender systems can recommend to the user i those items having the top filled/predicted values in the row $\boldsymbol{X}_{i,:}$. One common approach to CF is to approximate $\boldsymbol{X}$ by a dense matrix $\boldsymbol{Y}$ whose entries can be explained by some common knowledge such as *latent factors*. Traditionally, CF methods focus on items in one domain. For example, a video-rental recommender system like Netflix may regard items as all its available videos. In practice, the rating matrix in a single domain can be too sparse to learn satisfactory latent factors (Su & Khoshgoftaar, 2009). Furthermore, with the popularity of e-commerce systems (e.g., Amazon, TripAdvisor, etc.) and social media (e.g., Facebook, Twitter, etc.), users can provide feedbacks to items in multiple domains (e.g., books, DVDs, music, electronics, etc. in Amazon). It is desirable to have a cross-domain recommender system that is able to recommend items in different domains to a user, helping cross-selling the products and/or discovering new customers.

The above motivates recent studies on the *Cross Domain Collaborative Filtering* (CDCF) (Li, 2011) problem: given d rating matrices $\boldsymbol{X}^{(k)} \in \mathbb{R}^{n^{(k)} \times m^{(k)}}$, $1 \leq k \leq d$, each corresponding to $n^{(k)}$ users[1] and $m^{(k)}$ items in a specific domain k, fill the blanks in $\boldsymbol{X}^{(k)}$'s jointly. The CDCF methods are expected to improve the quality of recommendations because blanks in each $\boldsymbol{X}^{(k)}$ can be predicted based on not only the knowledge/latent factors in that domain, but also the *correlated knowledge transferred from other domains*. The correlation is less obscured by the sparsity since it can be inferred from multiple $\boldsymbol{X}^{(k)}$'s. Furthermore, with plenty domains available in an e-commerce system and social media, CDCF methods are also expected to further improve the quality of recommendations as d increases.

However, we find that in practice, existing CDCF methods usually give limited improvement in performance, and sometimes, adding new domains even results in *worse* performance. This is mainly due to how the correlation between domains is captured. To transfer knowledge between domains, most existing CDCF methods employ a

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

[1] Users in different domains can overlap.

technique called the *sub-structure sharing* which factorizes each $\boldsymbol{Y}^{(k)}$ into multiple matrices and in the meanwhile requires some of them, denoted as $\mathcal{S}^{(k)}$, to be shared by all domains. For example, it is common to tri-factorize each $\boldsymbol{Y}^{(k)} \in \mathbb{R}^{n^{(k)} \times m^{(k)}}$ into $\boldsymbol{Y}^{(k)} = \boldsymbol{U}^{(k)}\boldsymbol{B}^{(k)}\boldsymbol{V}^{(k)\top}$, where $\boldsymbol{U}^{(k)} \in \mathbb{R}^{n^{(k)} \times p^{(k)}}$ is a user-cluster matrix whose each entry $\boldsymbol{U}^{(k)}_{i,s}$ denotes how much the user i belongs to the user-cluster/latent factor s, $\boldsymbol{V}^{(k)} \in \mathbb{R}^{m^{(k)} \times q^{(k)}}$ is an item-cluster matrix whose each entry $\boldsymbol{V}^{(k)}_{j,t}$ denotes how much the item j is assigned to the item-cluster t, and $\boldsymbol{B}^{(k)} \in \mathbb{R}^{p^{(k)} \times q^{(k)}}$ a *rating pattern* matrix (or codebook) describing the linear relationship between clusters in $\boldsymbol{U}^{(k)}$ and $\boldsymbol{V}^{(k)}$. Then, to transfer knowledge, one can either share the $\mathcal{S}^{(k)} = \{\boldsymbol{U}, \boldsymbol{V}\}$ (Pan et al., 2010) or $\mathcal{S}^{(k)} = \boldsymbol{B}$ (Gao et al., 2013b;a; Li et al., 2009; Long et al., 2012) across all domains. The shared $\mathcal{S}^{(k)}$ captures the correlation between domains and is used as a *bridge* via which the knowledge in $\boldsymbol{Y}^{(k)} \backslash \mathcal{S}^{(k)}$ can be transferred. But sharing $\mathcal{S}^{(k)}$ also create linear dependency between all $\boldsymbol{Y}^{(k)} \backslash \mathcal{S}^{(k)}$'s. Consider $\mathcal{S}^{(k)} = \{\boldsymbol{U}, \boldsymbol{V}\}$. For any two domains k and l, we have $\boldsymbol{Y}^{(k)} = \boldsymbol{U}\boldsymbol{B}^{(k)}\boldsymbol{V}^{\top}$ and $\boldsymbol{Y}^{(l)} = \boldsymbol{U}\boldsymbol{B}^{(l)}\boldsymbol{V}^{\top}$. By $\boldsymbol{Y}^{(k)}\boldsymbol{V}(\boldsymbol{B}^{(k)})^{-1} = \boldsymbol{U} = \boldsymbol{Y}^{(l)}\boldsymbol{V}(\boldsymbol{B}^{(l)})^{-1}$, we obtain $\boldsymbol{B}^{(k)} = \boldsymbol{L}\boldsymbol{B}^{(l)}$ for some $\boldsymbol{L} \in \mathbb{R}^{p \times p}$. That is, $\boldsymbol{Y}^{(k)} \backslash \mathcal{S}^{(k)} = \boldsymbol{B}^{(k)}$ and $\boldsymbol{Y}^{(l)} \backslash \mathcal{S}^{(l)} = \boldsymbol{B}^{(l)}$, which denote the knowledge to be transferred, are linearly dependent with each other. This linear dependency exists in most current CDCF methods. Therefore, only the *linearly correlated knowledge* between domains can be identified and transferred.

In the real world, the correlation of knowledge between domains is usually non-linear. For example, consider two domains "books" and "movies." Suppose the ratings in $\boldsymbol{X}^{(movies)}$ in the "movies" domain can be explained by (but not limited to) a latent factor $f^{(movies)}_{hit}$ denoting the "box office hit," and the ratings in $\boldsymbol{X}^{(books)}$ in the "books" domain can be explained by two latent factors $f^{(books)}_{visibility}$ and $f^{(books)}_{curiosity}$ representing the "visibility of the book" and "user's curiosity about the story" respectively. The $f^{(movies)}_{hit}$ and $f^{(books)}_{visibility}$ may be linearly correlated, as the more a movie adapted from a book is played, the more the original book is visible to users. If so, these two latent factors and their correlation will be captured by $\boldsymbol{Y}^{(movies)}$ and $\boldsymbol{Y}^{(books)}$ using existing CDCF methods. Nevertheless, it is likely that the $f^{(movies)}_{hit}$ and $f^{(books)}_{curiosity}$ are non-linearly correlated, as users may be highly curious about the story in the original book either when the box office record of the adapted movie is high due to the public praise of screenplay, or when the box office record is low due to the criticism of unfaithful adaptation. In this case, the factor $f^{(books)}_{curiosity}$ will be missed by existing CDCF methods, resulting in degraded $\boldsymbol{Y}^{(books)}$.

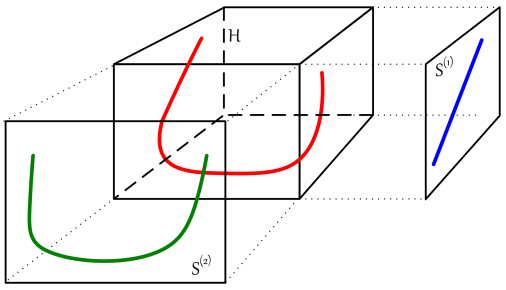

Figure 1. Different $\boldsymbol{Y}^{(k)}$'s share the projections of a hyper-structure $\mathcal{H}$. So, a pattern (e.g., the blue line) in some projection can be non-linearly dependent with that of another (e.g., the green curve).

In this paper, we propose a new knowledge transfer technique for CDCF, called the *Hyper-Structure Transfer* (HST), that captures the *non-linear* correlation of knowledge between domains. As in the sub-structure sharing, HST factorizes each $\boldsymbol{Y}^{(k)}$ into multiple matrices. But instead of fixing $\mathcal{S}^{(k)}$ across domains, HST requires $\mathcal{S}^{(k)}$ in each domain to be a *projection* of some more complex structure $\mathcal{H}$, called the *hyper-structure*, shared between domains. HST enables the knowledge transfer by using $\mathcal{H}$ as a bridge. Furthermore, since $\mathcal{S}^{(k)}$'s of different domains are projections of a more complex structure, they can be non-linearly dependent with each other, and *so are the domain-specific matrices* $\boldsymbol{Y}^{(k)} \backslash \mathcal{S}^{(k)}$'s. Figure 1 shows an example. Hence, HST allows the non-linearly correlated knowledge between domains to be identified and transferred.

HST brings new challenges: how to define $\mathcal{H}$ and the projections? And how to solve $\boldsymbol{Y}^{(k)}$'s (and $\mathcal{H}$) using the sparse ratings in $\boldsymbol{X}^{(k)}$'s? We study these problems in depth and propose an HST model, called the *Minimal Orthogonal Tensor Approximation with Residuals* (MOTAR) that can accommodate arbitrarily complex hyper-structure in the tensor form. We then show that the MOTAR objective can be transformed into a Canonical Polyadic (CP) decomposition problem (Kiers et al., 1999; Kolda & Bader, 2009; Bader & Kolda, 2006) of tensors and solved using existing techniques. Extensive experiments are conducted on both real and synthetic datasets and the results show that MOTAR can achieve up to 50% improvement in prediction accuracy as compared with the state-of-the-art CDCF methods. In addition, MOTAR always give better performance when taking into account new domains (i.e., increasing d).

Further Related Work. In addition to the studies mentioned above, there are CDCF methods (Cao et al., 2010; Shi et al., 2011; Zhang et al., 2012) that model $\boldsymbol{Y}^{(k)}$ using random variables. In these models, some random variables are assumed to be identical in distribution across domains, thus creating linear dependency, in probability, between the rest variables.

To exploit the data in large recommender systems (such as Amazon and Facebook) where users are allowed to rate different items in different domains/categories, some extensions of the above factorization methods are proposed, including the TrAnsfer Learning for MUltiple Domains (TALMUD) (Moreno et al., 2012), the extended Cluster-level Latent Factor Model (CLFM) (GAO ET AL., 2013B;A) and the Cross-Domain Triadic Factorization (CDTF) (Hu et al., 2013). These work target $d > 2$.

In this paper, we focus on CDCF across multiple domains ($d > 2$). We do not assume the availability of any side information (e.g., tags (Chen et al., 2013), implicit feedbacks (Moreno et al., 2012), etc.). We also make no assumption about the distributions of latent representation of users and items as in (Cao et al., 2010; Shi et al., 2011; Zhang et al., 2012).

2. Challenges and Evidence

In this section, we give evidence of limitations of existing CDCF methods.

2.1. Linear Knowledge Transfer

Despite the correlation of knowledge between domains is usually non-linear, existing CDCF methods can only transfer the linearly correlated knowledge between domains. Thus, they may give little help or even result in *negative transfer* (Rosenstein et al., 2005), where the availability of new domains degrades performance. Traditional studies (Argyriou et al., 2008; Bakker & Heskes, 2003; Ben-David & Schuller, 2003) aim to lower the chance of negative transfer by 1) grouping the domains such that the domains in a group are more likely to have transferable knowledge, and then 2) performing separate learning task for each group. However, this significantly limits the data available in each group, and the transferability of knowledge inside each group can still be limited by the linearity.

To give a clear case of the above limitation, we conduction experiments using DBLP citation dataset (Tang et al., 2008) and MovieLens rating dataset[2] to demonstrate the non-linearity of the correlation between domains. We measure the non-linearity by (1) tri-factorizing each $\boldsymbol{Y}^{(k)}$ into $\boldsymbol{Y}^{(k)} = \boldsymbol{U}^{(k)}\boldsymbol{B}^{(k)}\boldsymbol{V}^{(k)\top}$ independently to obtain the overall approximation score $a = \sum_{k=1}^{d} \|\boldsymbol{X}^{(k)} - \boldsymbol{U}^{(k)}\boldsymbol{B}^{(k)}\boldsymbol{V}^{(k)\top}\|_{\mathcal{F}}^2$ where $\|\cdot\|_{\mathcal{F}}$ denotes the Frobenius norm, and (2) tri-factorizing $\boldsymbol{Y}^{(k)}$'s jointly by sharing $\boldsymbol{B}^{(k)} = \boldsymbol{B}$ to obtain another approximation score b, and (3) derive a ratio, named *Non-Linearity Ratio* (NLR), by $(b - a)/b$. The larger the NLR, the more approximation error is introduced due to the shared linear explanation $\boldsymbol{B}$ in step (2), implying that the correlation between datasets tends to be zero or non-linear.

The NLRs of domain 1 over different domain pairs[3] in the DBLP and MovieLens datasets are summarized in Table 1. As we can see, the NLRs are approximately between $[0.34 \sim 0.5]$ in all domain pairs, which are very high.

2.2. Maladaptive Bridge

The above limitation can be amplified by a suboptimal bridge. Existing CDCF methods across multiple domains ($d > 2$) rely on either the *pairwise* or *all-intersectional* extensions of the sub-structure sharing technique. In the former (e.g, TALMUD (Moreno et al., 2012)), each pair $(\boldsymbol{Y}^{(k)}, \boldsymbol{Y}^{(l)})$ shares a bridge that can be different from that of the other pairs; while in the latter (e.g., CDTF (Hu et al., 2013) and extended CLFM (Gao et al., 2013a)), all $\boldsymbol{Y}^{(k)}$'s share the same bridge. We argue that *neither* of the bridges in these two extensions is good enough to capture the correlation between multiple domains. Consider four domains "books," "movies," "music," and "electronics." Suppose there is a pattern that users who have watched a movie and listened to its soundtrack (showing interests in related products) are usually welling to buy its original book also. Note that this pattern involves only partial domains (no "electronics"), and it *cannot* be captured by the above bridges, no matter how dense the rating matrices $\boldsymbol{X}^{(k)}$'s are, because the bridges of pairwise extensions are *too weak* to capture any correlation between more than two domains, yet the bridge of all-intersectional extensions are *too strong* to capture any correlation between partial domains. The strength of the bridge should be *adaptive* to capture the correlation between *any domain combination*.

To show the impacts of maladaptive bridges to performance, we test the error rates of existing CDCF extensions to multiples domains against different d's, as shown in Figures 4 and 5. In TALMUD, the decrease in error rate marginalizes quickly as d grows. Since only the correlation between domain pairs can be captured, this shows

[2] http://grouplens.org/datasets/movielens/.

[3] The DBLP citation dataset contains 180,640 authors (users), 141,507 papers (items) and 1,495,081 citation relations. Each paper is associated with authors, year, venue, and citations. We use the categories listed in Microsoft Academic Search to divide the venues into different domains: Algorithm & Theory (A&T), Data Mining (DM), Databases (DB), Distribution & Parallel Computing (D&PC), Human-Computer Inter- action (HCI), Machine Learning & Pattern Recognition (ML&PR), Artificial Intelligence (AI), Natural Language & Speech (NL&S), Programming Language (PL), Software Engineering (SE), World Wide Web (WWW). The MovieLens rating dataset contains 69,878 users, 10,677 movies (items), and 10,000,054 ratings. We categorize the movies into the following domains: Action (ACT), Adventure (ADV), Animation (ANI), Children (CHI), Comedy (COM), Crime (CRI), Drama (DRA), Horror (HOR), Sci-Fi (S&F), and Thriller (THR).

that there are many patterns that involve more than two domains, and these patterns are missed during the knowledge transfer. We can also see that the all-intersectional methods (CDTF and extended CLFM) may lead to the negative transfer when adding new domains. Since these methods only capture the correlation between all domains, this shows that a domain can be correlated to only parts of the rest domains. So, adding a new domain prevents the all-intersectional correlation between the rest domains from being captured.

Given the above limitations, it is crucial to have a new CDCF technique that is able to transfer non-linearly correlated knowledge between domains, via an adaptive bridge.

3. Hyper-Structure Transfer

In this section, we propose the notion of Hyper-Structure Transfer (HST) and its models that allows the non-linearly correlated knowledge between domains to be transferred.

3.1. General Model

A general model of Hyper-Structure Transfer (HST) can be written as:

$$\begin{gathered}\arg\min_{\boldsymbol{U}^{(k)},\boldsymbol{V}^{(k)},\mathcal{H}} \mathcal{L}\left(\boldsymbol{U}^{(k)},\boldsymbol{V}^{(k)},\mathcal{H}\right), \\ \mathcal{L}=\sum_{k=1}^{d}\left\|\left(\boldsymbol{X}^{(k)}-\boldsymbol{U}^{(k)}\times \operatorname{proj}^{(k)}(\mathcal{H})\times \boldsymbol{V}^{(k)\top}\right)\circ \boldsymbol{W}^{(k)}\right\|_{\mathcal{F}}^{2},\end{gathered} \tag{1}$$

subject to constraints $\boldsymbol{U}^{(k)} \geq \boldsymbol{O}$, $\boldsymbol{U}^{(k)}\mathbf{1}=\mathbf{1}$, $\boldsymbol{V}^{(k)} \geq \boldsymbol{O}$, and $\boldsymbol{V}^{(k)}\mathbf{1}=\mathbf{1}$, where d is the number of participating domains, $\boldsymbol{X}^{(k)} \in \mathbb{R}^{n^{(k)}\times m^{(k)}}$ is the rating matrix to approximate, $\boldsymbol{U}^{(k)} \in \mathbb{R}^{n^{(k)}\times p^{(k)}}$ is the user-cluster matrix whose each entry $u_{i,s}^{(k)}$ denote how much the user i belong to the cluster/latent factor s, $\boldsymbol{V}^{(k)} \in \mathbb{R}^{m^{(k)}\times q^{(k)}}$ is the item-cluster matrix, $\mathcal{H}$ is a high-order structure, called the *hyper-structure*, whose dimension is higher than $p^{(k)}\times q^{(k)}$ for all k, and $\operatorname{proj}^{(k)}(\cdot)$ is some projection map onto $\mathbb{R}^{p^{(k)}\times q^{(k)}}$, $\circ$ is the entry-wise product, and $\boldsymbol{W}^{(k)} \in \mathbb{R}^{n^{(k)}\times m^{(k)}}$ is the indicator matrix of $\boldsymbol{X}^{(k)}$ whose each entry is defined as:

$$\boldsymbol{W}_{ij}^{(k)}=\begin{cases}1, & \text{if } \boldsymbol{X}_{ij}^{(k)}\neq 0,\\ 0, & \text{otherwise,}\end{cases}$$

and $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm. Note that the hyper-structure $\mathcal{H}$ is fixed across domains. Basically, HST approximates each $\boldsymbol{X}^{(k)}$ by a dense $\boldsymbol{Y}^{(k)}=\boldsymbol{U}^{(k)}\times \operatorname{proj}^{(k)}(\mathcal{H})\times \boldsymbol{V}^{(k)\top}$ such that they look similar at the positions where the ratings are available. Similar to the existing methods we have seen in Section 1, each $\boldsymbol{Y}^{(k)}$ is tri-factorized into $\boldsymbol{U}^{(k)}\boldsymbol{B}^{(k)}\boldsymbol{V}^{(k)\top}$ for some rating pattern matrix $\boldsymbol{B}^{(k)} \in \mathbb{R}^{p^{(k)}\times q^{(k)}}$. But instead of fixing $\boldsymbol{B}^{(k)}=\boldsymbol{B}$ across domains, HST allows $\boldsymbol{B}^{(k)}$'s to be *different projections* of $\mathcal{H}$.

HST enables the knowledge transfer by using $\mathcal{H}$ as a bridge. Furthermore, the $\boldsymbol{B}^{(k)}$'s can capture the non-linearly correlation between domain. For example, suppose in domain k there is a rating pattern that can be depicted as a blue line in Figure 1. Suppose there is another pattern, a curve, in domain l. Traditional CDCF methods cannot capture these patterns due to their non-linear correlation, so any user/item clusters that can only be linked through these patterns will be missed. On the other hand, the correlation between these two patterns can be captured in HST by a more complex structure—a high-order curve. Therefore, HST can achieve better approximation.

The idea of learning from data in a space more complex than where the original data reside has appeared in the machine learning field. For example, a kernel machine (Schölkopf & Smola, 2002) employs a kernel function to lift the observed data instances (users/items) to a high-order space (called the feature space), and then performs the learning task in that space. However, HST differs from the kernel machines in that the target of the lifting is *not* observed (it is the unknown correlation between rating patterns to be lifted). Therefore, we cannot simply pass the correlation into the kernel function to obtain $\mathcal{H}$. The precise definitions of $\mathcal{H}$ and the projection function remains open.

3.2. MOTAR

The $\mathcal{H}$ and the projection function $\operatorname{proj}(\cdot)$ in Eq. (1) can be specified by domain experts if some priors knowledge about the correlation is available. However, it is hard to know all priors. Next, we present the *Minimal Orthogonal Tensor Approximation with Residuals* (MOTAR), a special case of HST that allows $\mathcal{H}$ to capture arbitrarily complex correlation in the tensor form without the need for priors. In MOTAR, the $\mathcal{H}$ and $\operatorname{proj}(\cdot)$ are defined as

$$\mathcal{H}=\boldsymbol{\mathcal{B}} \in \mathbb{R}^{p^{(1)}\times q^{(1)}\times\cdots\times p^{(d)}\times q^{(d)}} \tag{2}$$

and

$$\operatorname{proj}^{(k)}(\boldsymbol{\mathcal{B}})_{s,t}=\sum_{i^{(l)}=1,j^{(l)}=1,\forall l\neq k}^{p^{(l)},q^{(l)}} \boldsymbol{\mathcal{B}}_{i^{(1)},j^{(1)},\cdots,s,t,\cdots,i^{(d)},j^{(d)}} \tag{3}$$

respectively. Basically, MOTAR assumes that the hyper-structure $\mathcal{H}$ is a tensor $\boldsymbol{\mathcal{B}}$ with $2d$ modes, and each $\operatorname{proj}^{(k)}(\boldsymbol{\mathcal{B}}) \in \mathbb{R}^{p^{(k)}\times q^{(k)}}$ is a projection of $\boldsymbol{\mathcal{B}}$ onto some two modes. The projections are orthogonal to each other as they target distinct modes. By definition of $\operatorname{proj}(\cdot)$ in Eq. (3), $2d$ is the minimum number of modes that allows $\boldsymbol{\mathcal{B}}$ to be orthogonally projected onto the rating pattern matrices in different domains, yet $\boldsymbol{\mathcal{B}}$ is large enough to capture any correlation (in tensor form) of *arbitrarily complexity*:

Theorem 1. *Let $\boldsymbol{\mathcal{C}}$ be a tensor with c modes, $c>2d$, that denotes the correlation between the ratting patterns of dif-*

ferent domains. MOTAR can capture $\mathcal{C}$ by letting

$$\mathcal{B}_{s^{(1)},t^{(1)},\cdots,s^{(d)},t^{(d)}} = \sum_{i^{(1)},\cdots,i^{(c-2d)}} \mathcal{C}_{s^{(1)},t^{(1)},\cdots,s^{(d)},t^{(d)},i^{(1)},\cdots,i^{(c-2d)}}.$$

The above theorem can be easily proved by showing that $\text{proj}^{(k)}(\mathcal{B}) = \text{proj}^{(k)}(\mathcal{C})$.

3.3. Residuals for Adaptive Bridge

When multiple domains are available ($d > 2$), a correlation pattern may involve only partial domains (an example is given in Section 2.2). The bridge for knowledge transfer needs to be *adaptive* to capture the patterns involving arbitrary domain combinations.

MOTAR has another advantage in that the bridge $\mathcal{B}$ defined in Eq. (2) can be readily extended to be adaptive, by creating one extra dimension in each mode, i.e.,

$$\mathcal{B} \in \mathbb{R}^{(p^{(1)}+1)\times(q^{(1)}+1)\times\cdots\times(p^{(d)}+1)\times(q^{(d)}+1)}, \quad (4)$$

such that each the rating pattern matrix $\text{proj}^{(k)}(\mathcal{B}) \in \mathbb{R}^{(p^{(k)}+1)\times(q^{(k)}+1)}$ relates one extra user-cluster and one extra item-cluster, called the *residual clusters*. MOTAR requires the columns corresponding to the residual clusters in the user-cluster and item-cluster matrices to be $\mathbf{0}$, i.e.,

$$\boldsymbol{Y}^{(k)} = \left[\boldsymbol{U}^{(k)}, \mathbf{0}\right] \text{proj}^{(k)}(\mathcal{B}) \left[\boldsymbol{V}^{(k)}, \mathbf{0}\right]^\top,$$

so the values of $\boldsymbol{Y}^{(k)}$ will *not* be affected by the extension of $\text{proj}^{(k)}(\mathcal{B})$. This allows the new values in $\text{proj}^{(k)}(\mathcal{B})$ to reflect some partial structure of $\mathcal{B}$ that *only affects the other domains*. In other words, the residual clusters in each domain denote "the latent factors which cannot be explained by $\boldsymbol{X}^{(k)}$ in this domain." Therefore, the bridge $\mathcal{B}$, which links any combination of the user-clusters, item-clusters, *and their negation* across different domains, is adaptive.

Finally, we obtain the objective of MOTAR as follows:

$$\begin{gathered}\arg\min_{\boldsymbol{U}^{(k)},\boldsymbol{V}^{(k)},\mathcal{B}} \mathcal{L}\left(\boldsymbol{U}^{(k)}, \boldsymbol{V}^{(k)}, \mathcal{B}\right), \\ \mathcal{L} = \sum_{k=1}^{d} \left\| \left(\boldsymbol{X}^{(k)} - \left[\boldsymbol{U}^{(k)}, \mathbf{0}\right] \text{proj}^{(k)}(\mathcal{B}) \left[\boldsymbol{V}^{(k)}, \mathbf{0}\right]^\top\right) \circ \boldsymbol{W}^{(k)} \right\|_{\mathcal{F}}^2,\end{gathered} \quad (5)$$

subject to constraints $\boldsymbol{U}^{(k)} \geq \boldsymbol{O}$, $\boldsymbol{U}^{(k)}\mathbf{1} = \mathbf{1}$, $\boldsymbol{V}^{(k)} \geq \boldsymbol{O}$, and $\boldsymbol{V}^{(k)}\mathbf{1} = \mathbf{1}$. Note that the l_1 normalization constraints on each row of $\boldsymbol{U}^{(k)}$ and $\boldsymbol{V}^{(k)}$ keeps the objective well-defined (Gu et al., 2011).

4. Objective Solving

One way to solve Eq. (5) is to randomly initialize $\mathcal{B}$, and repeat until convergence the process of updating $\mathcal{B}$ for each domain k such that the $\text{proj}^{(k)}(\mathcal{B})$ lead to a better approximation. However, this approach is very inefficient (either in space and time, and may even be infeasible) due to the *huge* number of variables to solve in $\mathcal{B}$ in each iteration.

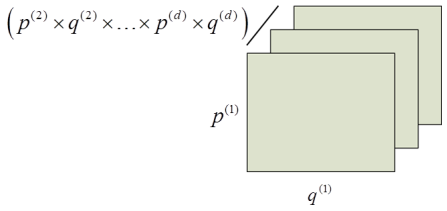

Figure 2. An example cubicization of $\mathcal{B}$ over domain 1.

Instead of updating $\mathcal{B}$ directly in the above iterating process, we propose updating $\text{proj}^{(k)}(\mathcal{B})$ for each domain alternately to improve the overall approximation. This can significantly reduces the number of variables to solve for each update (from $\Pi_l p^{(l)} q^{(l)}$ to $p^{(k)} q^{(k)}$). However, it creates another challenge: after updating $\text{proj}^{(k)}(\mathcal{B})$ in the current iteration, how to obtain $\text{proj}^{(l)}(\mathcal{B})$ to be updated in the next iteration? We link $\text{proj}^{(k)}(\mathcal{B})$ and $\text{proj}^{(l)}(\mathcal{B})$ based on a key observation, as described below.

Let's first define a *cubicization* of $\mathcal{B}$ over some domain k, denoted by $\tilde{\mathcal{B}}^{(k)} \in \mathbb{R}^{p^{(k)} \times q^{(k)} \times \Pi_{l\neq k} p^{(l)} q^{(l)}}$, a 3-mode tensor whose the first and second modes are the modes of domain k in $\mathcal{B}$ and the third mode is the concatenation of the rest modes in $\mathcal{B}$, as sown in Figure 2. Note that

$$\text{proj}^{(k)}(\mathcal{B}) = \sum_h \tilde{\mathcal{B}}^{(k)}_{:,:,h},$$

where $\tilde{\mathcal{B}}^{(k)}_{:,:,h} \in \mathbb{R}^{p^{(k)} \times q^{(k)}}$ denotes the h-th slice of $\tilde{\mathcal{B}}^{(k)}$ along the third mode. By CP decomposition, we have $\tilde{\mathcal{B}}^{(k)} = \sum_{r=1}^{z} \boldsymbol{a}_r \otimes \boldsymbol{e}_r \otimes \boldsymbol{c}_r$ for some $\boldsymbol{a}_r \in \mathbb{R}^{p^{(k)}}$, $\boldsymbol{e}_r \in \mathbb{R}^{q^{(k)}}$, $\boldsymbol{c}_r \in \mathbb{R}^{\Pi_{l\neq k} p^{(l)} q^{(l)}}$ and hyperparameter z, as shown in Figure 3. Let $\boldsymbol{A}^{(k)} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_z] \in \mathbb{R}^{n^{(k)} \times z}$, $\boldsymbol{E}^{(k)} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots] \in \mathbb{R}^{m^{(k)} \times z}$, and $\boldsymbol{C}^{(k)} = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots] \in \mathbb{R}^{\Pi_{l\neq k} p^{(l)} q^{(l)} \times z}$, we have $\tilde{\mathcal{B}}^{(k)}_{i,j,h} = \sum_{r=1}^{z} \boldsymbol{A}^{(k)}_{i,r} \boldsymbol{E}^{(k)}_{j,r} \boldsymbol{C}^{(k)}_{h,r}$, and we can write the h-th slice of $\tilde{\mathcal{B}}^{(k)}$ along the third mode as

$$\tilde{\mathcal{B}}^{(k)}_{:,:,h} = \boldsymbol{A}^{(k)} \boldsymbol{\Phi}^{(k)}_h \boldsymbol{E}^{(k)\top}, \quad (6)$$

where $\boldsymbol{\Phi}^{(k)}_h = diag(\boldsymbol{C}^{(k)}_{h,:}) \in \mathbb{R}^{z\times z}$ is a diagonal matrix. This leads to

$$\begin{aligned}\text{proj}^{(k)}(\boldsymbol{B}) &= \textstyle\sum_h \tilde{\mathcal{B}}^{(k)}_{:,:,h} \\ &= \textstyle\sum_h \boldsymbol{A}^{(k)} \boldsymbol{\Phi}^{(k)}_h \boldsymbol{E}^{(k)\top} \\ &= \boldsymbol{A}^{(k)} \boldsymbol{\Psi}^{(k)} \boldsymbol{E}^{(k)\top},\end{aligned} \quad (7)$$

where $\boldsymbol{\Psi}^{(k)} = \sum_h \boldsymbol{\Phi}^{(k)}_h$ is diagonal.

Eq. (7) implies that, during each iteration, we can simply update $\text{proj}^{(k)}(\boldsymbol{B})$ by updating $\boldsymbol{A}^{(k)}$ and $\boldsymbol{E}^{(k)}$. Furthermore, if we fix $\boldsymbol{\Psi}^{(k)}$ (and $\boldsymbol{\Phi}^{(k)}_h$, $\forall h$), we can use Eq. (6) to

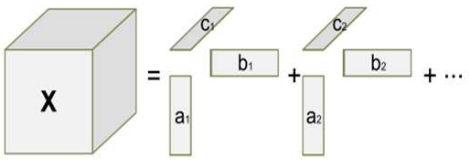

Figure 3. The CP decomposition of a three-order tensor

reconstruct $\mathcal{B}$ and then obtain the $\text{proj}^{(l)}(\mathcal{B})$ to be updated in the next iteration by using Eq. (7) again. The number of variables to solve in $\boldsymbol{A}^{(k)}$ and $\boldsymbol{E}^{(k)}$ is much smaller than that in $\mathcal{B}$.

With the above observation, we can employ the multiplicative gradient descent method (Ding et al., 2006) to update the $\tilde{\boldsymbol{U}}^{(k)} = [\boldsymbol{U}^{(k)}, \boldsymbol{0}]$, $\tilde{\boldsymbol{V}}^{(k)} = [\boldsymbol{V}^{(k)}, \boldsymbol{0}]$, $\boldsymbol{A}^{(k)}$, and $\boldsymbol{E}^{(k)}$ in each iteration using the update rules described in Section 1 of the supplementary material.

5. Experiment

In this section, we conduct experiments over both synthetic and real datasets to compare the performance of MOTAR with that of the state-of-the-art CDCF techniques.

Metric. We adopt a widely used metric for the CF problems—the Mean Absolute Error (MAE) (Su & Khoshgoftaar, 2009)—to evaluate the prediction quality of $\boldsymbol{Y}^{(k)}$

$$MAE(\boldsymbol{Y}^{(k)}; \boldsymbol{Z}^{(k)}) = \sum_{\boldsymbol{z}^{(k)}_{i,j} \text{ is not blank}} \frac{|\boldsymbol{Z}^{(k)}_{i,j} - \boldsymbol{Y}^{(k)}_{i,j}|}{\|\boldsymbol{Z}^{(k)}\|},$$

where $\boldsymbol{Z}^{(k)} \in \mathbb{R}^{n^{(k)} \times m^{(k)}}$ denotes the ratings that are held out from $\boldsymbol{X}^{(k)}$ during the training process, and $\|\boldsymbol{Z}^{(k)}\|$ is the number of ratings in $\boldsymbol{Z}^{(k)}$. The smaller the MAE, the higher the accuracy. We hold out data by time. Specifically, we cut the elder 90% ratings to $\boldsymbol{X}^{(k)}$ and the later 10% to $\boldsymbol{Z}^{(k)}$. And we tune the parameters by randomly selecting some ratings from $\boldsymbol{X}^{(k)}$ as the validation set.

Baselines. We compare MOTAR with the TALMUD (Moreno et al., 2012), CDTF (Hu et al., 2013) and CLFM (Gao et al., 2013a). Note that the CDTF assumes the users in different domains are identical, so we preprocess the datasets accordingly to allow the comparison. Generally, the objectives of in existing CDCF methods (including MOTAR) are *not* convex thus the solutions are subject to local minimum. We overcome this by randomly initializing the variables to solve 20 times and pick the resulting $\boldsymbol{Y}^{(k)}$'s that achieve the lowest objective score. These strategy works well for all the methods considered.

We employ two real datasets: the DBLP citation dataset and the MovieLens movie rating dataset.

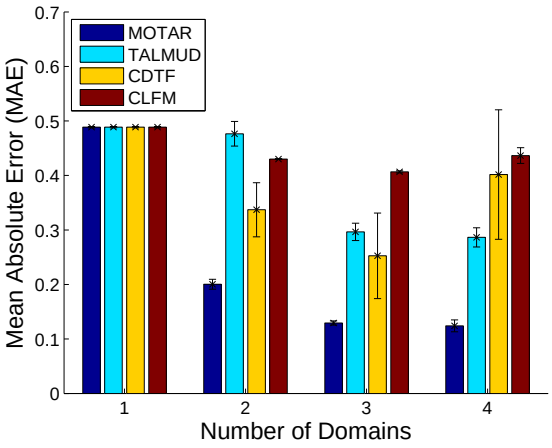

Figure 4. Performance comparison over multiple domains in the DBLP dataset.

DBLP Citation Dataset (Tang et al., 2008). The DBLP citation dataset contains 180,640 authors (users), 141,507 papers (items) from the year 1960 to 2010. We regard each rating $\boldsymbol{X}^{(k)}_{i,j}$ as 1 if the user i has ever cited the paper j. Based on this definition, there are 1,495,081 ratings in this datasets. We then partition these ratings into different $\boldsymbol{X}^{(k)}$'s by using the categories listed in the Microsoft Academic Search website.

MovieLens Rating Dataset.[4] The MovieLens rating dataset contains 69,878 users, 10,677 movies (items), and 10,000,054 ratings from the year 1970 to 2009. We use categories listed in its official website to divide the ratings into different domains. Some statistics of the above datasets are given in Table 1 of the supplementary material.

Note that some baseline methods such as CDTF require the users from different domains to be identical. So we prune users that give ratings in only a single domain (this setting is in favor of baselines). Also, we observe that users who have few publications/reviews tend to give ratings to random items. With these users, all algorithms give very poor performance because the test set contains random pulls mostly. Therefore, we also prune users who have few publications/reviews.

5.1. Results over Two Domains

Table 1 shows the MAEs of different CDCF methods over different domain pairs in the DBLP and MovieLens datasets. We randomly select 9 pairs of domains from each of the datasets respectively and report the MAEs in domain 1. The results show that MOTAR outperforms the baselines in almost all cases. The last column of

[4] http://grouplens.org/datasets/movielens/.

	Domain 1	Domain 2	NLR of domain 1	MOTAR	TALMUD	CDTF	CLFM
DBLP	A&T	DM	0.430629	**0.118268**	0.174754	0.181608	0.479545
	A&T	NL&S	0.439931	**0.132746**	0.252289	0.242860	0.488257
	DP&C	DB	0.411229	**0.116443**	0.388081	0.244012	0.537190
	D&PC	AI	0.394680	**0.114169**	0.428092	0.259215	0.541288
	HCI	DM	0.494192	**0.130390**	0.437423	0.316056	0.693805
	HCI	WWW	0.540556	**0.125405**	0.436462	0.280759	0.706178
	NL&S	AI	0.393302	**0.104989**	0.333125	0.189784	0.493210
	NL&S	ML&PR	0.379950	**0.090102**	0.291277	0.185565	0.491606
	NL&S	DB	0.413516	**0.088783**	0.356582	0.202629	0.498284
MovieLens	ACT	COM	0.364349	**0.316164**	0.327645	0.494919	0.597450
	ADV	THR	0.464158	**0.288821**	0.337397	0.501841	0.622482
	ANI	COM	0.516600	**0.306506**	0.395720	0.516925	0.623446
	CHI	ACT	0.540419	**0.304316**	0.440897	0.593867	0.635377
	COM	ACT	0.518781	**0.380733**	0.402398	0.638214	0.616576
	CRI	THR	0.344379	**0.275506**	0.332996	0.480467	0.536740
	DRA	THR	0.430730	**0.327201**	0.386229	0.591035	0.628440
	HOR	THR	0.445017	**0.254903**	0.385203	0.552813	0.590041
	S&F	THR	0.450324	**0.293761**	0.397893	0.579108	0.628371

Table 1. Performance comparison over two domains.

Table 1 shows the degrees of non-linearity (mentioned in Section 2) between different domain pairs. In general, the degree of non-linearity between each pair of domains are about 0.35~0.5, and we can see how the non-linearity affects the performance—the higher the degree of non-linearity, the more the MAE in the baseline methods. This shows that the performance of existing CDCF methods are largely limited by the sub-structure sharing technique, which can only transfer linearly correlated knowledge across domains. MOTAR, on the other hand, avoid this problem by sharing the hyper-structure.

5.2. Results over Multiple Domains

Now, we evaluate the performance of MOTAR over multiple domains ($d > 2$). We randomly select some pairs of domains in the DBLP and MovieLens datasets, then we run different algorithms and show their average MAEs as well as the standard deviation in Figures 4 and 5.

From the results, we can see that MOTAR significantly outperforms the other baselines in all cases due to the non-linear knowledge transfer. Note that when the number of domain is 1, all algorithms are reduced to the standalone tri-factorization, so all of them have the same error. Furthermore, we can also see that MOTAR gives more performance gain as the number d of domains increases. The more domains available, the more complex their correlation patterns can be. Thus, MOTAR has a higher chance to capture those patterns than the others when d is large.

Another important observation is that MOTAR does *not* tend to result in the negative knowledge transfer as we have discussed in Section 2.2. We can see from the figures that the MAE of MOTAR decreases monotonously as d increases. This is because that MOTAR employs an *adaptive* bridge that can capture the correlation patterns involving *any* domain combination.

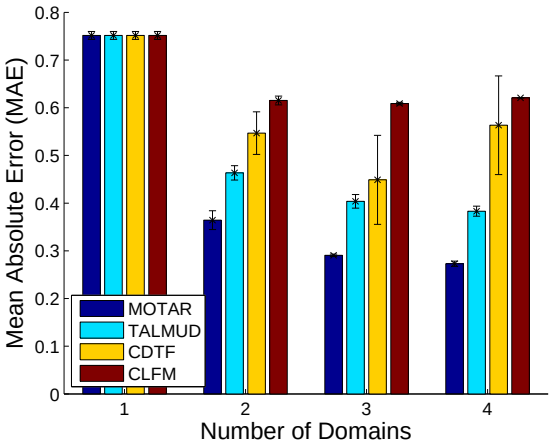

Figure 5. Performance comparison over multiple domains in the MovieLens dataset.

5.3. Effects of Non-linearity

To validate that MOTAR can capture arbitrarily complex correlations between domains, we create a synthetic dataset with known $\mathcal{B}$ in the ground truth to simulate the non-linear

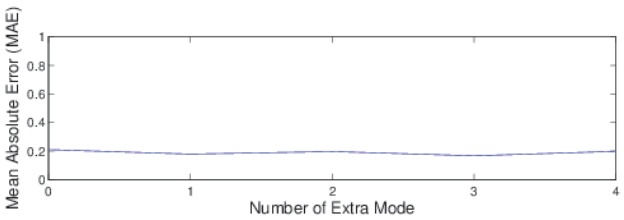

Figure 6. The effects of non-linearity.

condition. We then obtain the MAEs of MOTAR over this datatset by varying the number of modes of $\boldsymbol{\mathcal{B}}$ in the ground truth. We focus on two domains only to set aside the problem of maladaptive bridges (to be discussed in the supplementary material).

To create the dataset, we first define the number of users $n^{(k)}$ (resp. items $m^{(k)}$), the number of user-clusters $p^{(k)}$ (resp. item-clusters $q^{(k)}$) in k domains. And to control the non-linear correlation, we introduce extra modes in $\boldsymbol{\mathcal{B}}$. The number of the extra modes is denoted by exM and the number of dimensions in each extra modes is denoted by $r^{(l)}$, where $1 \leq l \leq exM$. We define the latent factors of users (resp. items) for each domain in ground truth, and randomly assign the latent factors to each user (resp. item) in the same domain.

Here we fix each $n^{(k)} = 20$, $m^{(1)} = 40$, $m^{(2)} = 50$ and each $p^{(k)} = q^{(k)} = r^{(k)} = 6$, and varies exM to simulate the non-linear complexity. And we generate the rating matrices by using $\boldsymbol{X}^{(k)} = \boldsymbol{U}^{(k)} \operatorname{proj}^{(k)} (\boldsymbol{\mathcal{B}}) \boldsymbol{V}^{(k)}$. Figure. 6 shows the accuracy achieved by MOTAR model under different degree of the non-linearity. As we can see, the performance of MOTAR is unaffected by the extra modes in $\boldsymbol{\mathcal{B}}$. This justifies that MOTAR can capture arbitrarily complex correlations between domains.

6. Conclusions and Future Work

We propose the notion of Hyper-Structure Transfer (HST) and its model called the Minimal Orthogonal Tensor Approximation with Residuals (MOTAR) that transfers non-linearly correlated knowledge between domains in the tasks of Cross-Domain Collaborative Filtering (CDCF). To the best of our knowledge, this is the first work that enables non-linear knowledge transfer. Empirical results show that MOTAR can achieve up to 50% reduction in prediction error rate as compared with the state-of-the-art CDCF methods. In addition, MOTAR always give better performance when taking into account new domains.

HST opens up numerous research directions in the future. First, there are many other ways to select $\mathcal{H}$ and the projection function $\operatorname{proj}(\cdot)$ in Eq. (1). For example, $\mathcal{H}$ could be some manifold in a high-order geometric space and the $\operatorname{proj}^{(k)}(\mathcal{H})$'s could be its local patches (overlapping with each other). Second, one can develop new regularizers for HST/MOTAR that best suite the targeted recommender systems where some particular side information (e.g., tags, implicit feedbacks, etc.) is available. Third, the in-depth studies about the non-linear correlation between domain knowledge will be valuable. Theoretical guarantee about the non-linear transferability wold be important too. Last but not the least, we find that the CP decomposition is the major performance bottleneck when solving the MOTAR objectives (despite that the training Algorithm shown in the supplementary material converges after only tens of outer-iterations in most cases). Alternatives to CP decomposition will be valuable for large-scale settings.

References

Argyriou, A., Maurer, A., and Pontil, M. An algorithm for transfer learning in a heterogeneous environment. In *Proc. of ECML PKDD*, 2008.

Bader, Brett W. and Kolda, Tamara G. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Trans. on Mathematical Software*, 32:635–653, 2006.

Bakker, Bart and Heskes, Tom. Task clustering and gating for bayesian multitask learning. *The Journal of Machine Learning Research*, 4:83–99, 2003.

Ben-David, S. and Schuller, R. Exploiting task relatedness for multiple task learning. In *Proc. of COLT/Kernel*, 2003.

Cao, Bin, Liu, Nathan N, and Yang, Qiang. Transfer learning for collective link prediction in multiple heterogenous domains. In *Proc. of ICML*, 2010.

Chen, Wei, Hsu, Wynne, and Lee, Mong Li. Making recommendations from multiple domains. In *Proc. of KDD*, 2013.

Ding, Chris, Li, Tao, Peng, Wei, and Park, Haesun. Orthogonal nonnegative matrix t-factorizations for clustering. In *Proc. of KDD*, 2006.

Gao, Sheng, Luo, Hao, Chen, Da, Li, Shantao, Gallinari, Patrick, and Guo, Jun. Cross-domain recommendation via cluster-level latent factor model. In *Proc. of ECML PKDD*, 2013a.

Gao, Sheng, Luo, Hao, Chen, Da, Li, Shantao, Gallinari, Patrick, Ma, Zhanyu, and Guo, Jun. A cross-domain recommendation model for cyber-physical systems. *IEEE Trans. on Emerging Topics in Computing*, 1:384–393, 2013b.

Gu, Quanquan, Ding, Chris, and Han, Jiawei. On trivial solution and scale transfer problems in graph regularized nmf. In *Proc. of IJCAI*, 2011.

Hofmann, Thomas. Latent semantic models for collaborative filtering. *ACM Trans. on Information Systems*, 22: 89–115, 2004.

Hu, Liang, Cao, Jian, Xu, Guandong, Cao, Longbing, Gu, Zhiping, and Zhu, Can. Personalized recommendation via cross-domain triadic factorization. In *Proc. of WWW*, 2013.

Hu, Yifan, Koren, Yehuda, and Volinsky, Chris. Collaborative filtering for implicit feedback datasets. In *Proc. of ICDM*, 2008.

Kiers, Henk AL, Ten Berge, Jos MF, and Bro, Rasmus. Parafac2-part i. a direct fitting algorithm for the parafac2 model. *Journal of Chemometrics*, 13:275–294, 1999.

Kolda, Tamara G and Bader, Brett W. Tensor decompositions and applications. *SIAM review*, 51:455–500, 2009.

Koren, Yehuda, Bell, Robert, and Volinsky, Chris. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, 2009.

Li, Bin. Cross-domain collaborative filtering: A brief survey. In *Proc. of ICTAI*, 2011.

Li, Bin, Yang, Qiang, and Xue, Xiangyang. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In *Proc. of IJCAI*, 2009.

Long, Mingsheng, Wang, Jianmin, Ding, Guiguang, Shen, Dou, and Yang, Qiang. Transfer learning with graph co-regularization. In *Proc. of AAAI*, 2012.

Moreno, Orly, Shapira, Bracha, Rokach, Lior, and Shani, Guy. Talmud: transfer learning for multiple domains. In *Proc. of CIKM*, 2012.

Pan, Weike, Xiang, Evan Wei, Liu, Nathan Nan, and Yang, Qiang. Transfer learning in collaborative filtering for sparsity reduction. In *Proc. of AAAI*, 2010.

Rosenstein, Michael T, Marx, Zvika, Kaelbling, Leslie Pack, and Dietterich, Thomas G. To transfer or not to transfer. In *Proc. of NIPS*, 2005.

Schölkopf, Bernhard and Smola, Alexander J. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

Shi, Yue, Larson, Martha, and Hanjalic, Alan. Tags as bridges between domains: Improving recommendation with tag-induced cross-domain collaborative filtering. *User Modeling, Adaption and Personalization*, 6787: 305–316, 2011.

Su, Xiaoyuan and Khoshgoftaar, Taghi M. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4, 2009.

Tang, Jie, Zhang, Jing, Yao, Limin, Li, Juanzi, Zhang, Li, and Su, Zhong. Arnetminer: extraction and mining of academic social networks. In *Proc. of KDD*, 2008.

Zhang, Yu, Cao, Bin, and Yeung, Dit-Yan. Multi-domain collaborative filtering. *arXiv preprint arXiv:1203.3535*, 2012.