# Multi-metric Learning For Cross-domain Recommendation

**Submitted for Blind Review**

**Abstract**

How to discover potential users of new released products is an important task in marketing campaigns, especially for companies have many established users. Indeed, transfer learning is a natural choice for addressing such problem, which can leverage the user data from historical products (i.e., auxiliary domains) for recommendation of new product (i.e., target domian). However, although most of the existing solutions of transfer learning work well on predicting user ratings, few of them can be directly exploited for the top-$k$ recommendation with respect to user preferences. To this end, in this paper, we propose a novel transfer learning approach based on multi-metric learning to address the problem of top-$k$ potential user recommendation. Specifically, by modeling the user behaviors on historical products, we first propose to learn a global distance metric for capturing the commonality of users and segmenting users into different neighborhoods. Furthermore, to enhance the local discriminability among different neighborhoods, we also learn some local distance metrics for better estimating the user preferences. With such multi-metric, a novel method is developed for recommending top-$k$ potential users of new products. Finally, we perform extensive experiments on real-world datasets, and the results clearly validate the effectiveness of the proposed approaches.

## Introduction

Discovering potential users of new released products plays an important role in marketing campaigns, especially for companies have many established products and users. For example, it is urgent for Apple to know how many of its established customers would like to buy the new released iPhone 5s. Intutively, one of the straightforward solutions is to directly recommend all the users of historical products to the new product. However, according to our real-world observations, users often have different forms of acceptance to the different products. Therefore, it is appealing to estimate users' personal preferences when making recommendation.

Indeed, the problem of potential user recommendation can be regarded as a cross-domain problem, where the historical products represent auxiliary domains and the new product represents target domain, respectively. In previous literatures, transfer learning based techniques is widely used for addressing such cross-domain recommendation problem (Pan and Yang (2010)). Among these techniques, metric learning is a good candidate for our problem because of its heterogeneous assumption of the feature dimensions. Specifically, metric learning solves the domain adaptation problem with performing the learning task in source domain and applying the learner to the target domain which is governed by a different data distributions. Recently, it has been widely applied to problems such as music recommendation (McFee, Barrington, and Lanckriet (2012)), webpage archiving (Law et al. (2012)) and partitioning (Lajugie, Arlot, and Bach (2013)). However, although traditional solutions of metric learning perform well on predicting user ratings, few of them can be directly exploited for the top-$k$ recommendation with respect to user preferences. Indeed, in real-world services, it is hard to obtain accurate user ratings, thus the service providers usually care more about the ability to extract top-$k$ users based on their preferences.

To this end, in this paper, we propose to develop a novel multi-metric learning approach to address the problem of top-$k$ potential user recommendation. However, there are several challenges along this line. First, **sparse connection**. Recommender systems assume that similar users like similar products, but this assumption does not hold in our scenario. This is because users have different acceptance for a newly launched product. Thus, the user connections between different products are sparse. In this paper, we learn several "local" distance metrics to address the sparse connection problem. Second, **behavior imbalance**. Users' similar behavior patterns are not always the reflection of their interests since the time schedule may be different, e.g., afternoon and middle-night. In this paper, we use the user behavior in different time schedules as features for better representing user interest. Third, **users skewness**. That is, not all users are need to be predicted. In fact, in our study, only less than 15% of all possible users are relevant to a marketing strategy for a company. Therefore, for the task of cross-domain metric learning in recommendation, we pay more attention to those users' records expressing cross-domain preference in the experiments.

To address the challenges mentioned above, in this paper, we propose a novel approach, namely Cross-domain Multi-metric Learning (CML), by combining both metric learning and active learning. Specifically, CML is a generative model that firstly captures the similarity of all the users by learning

a global distance metric. Then, CML further selects some representative users by active learning and forms several neighborhoods. Furthermore, to enhance the local discriminability among different neighborhoods, we also learn some local distance metrics for better estimating the user preferences. With such multi-metric, a novel method is developed for recommending top-$k$ potential users of new products. In addition, we define a new evaluation metric to verify the model's ability to extract top-$k$ target customers. Finally, we conduct empirical experimentation with real-world datasets to demonstrate the effectiveness of the proposed algorithm.

## Related work

We first review the previous work on distance metric learning. Then, we show a brief overview of active learning that are closely related to our learning method.

**Distance Metric Learning.** Most of the algorithms in this field aim to learn a distance metric from the side information which is typically presented in a set of pairs of "similar" or "dissimilar" objects. The optimal distance metric can be found by keeping "similar" objects close and separating "dissimilar" ones at the same time. In the past, Neighborhood Component Analysis(NCA) (Blitzer, Weinberger, and Saul (2005)) learns a distance metric with the nearest neighbor classifier which is extended to be large-margin nearest neighbor (LMNN) classifier (Weinberger and Saul (2009)). And alternative ways of solving the problem have been proposed (Nguyen and Guo (2008), Park et al. (2011), Der and Saul (2012)) . (Hong et al. (2011)) later proposed to learn a mixture of NCA metrics, while (Tarlow, Sutskever, and Zemel (2013)) generalized NCA to $k$-NN with $k > 1$.

Despite extensive development, most algorithms only depend on single distance metric which could be unreliable when the number of training samples is large. Furthermore, much previous work assumes randomly-selected data examples, which is not sufficient in identifying the optimal distance metric. Our proposed method aims to address these problems by multi-metric learning across different domains.

**Active Learning.** In many real-world problems, we encounter the difficulties when the unlabeled data are abundant but labeling data is expensive to obtain. Active learning for classification tasks has been widely studied (Wang et al. (2009); Tang et al. (2012)).

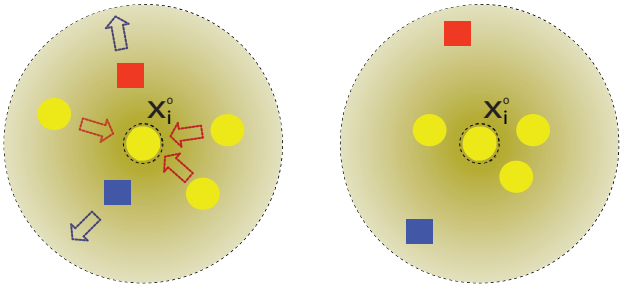
Compared with semi-supervised learning, active learning use methods to tackle the same problem from the opposite direction. While semi-supervised methods exploit what the learner thinks it knows about the unlabeled data, active methods attempt to explore the unknown aspects. It is therefore natural to think about combining the two. Some example formulations of semi-supervised active learning include (Bhagat et al. (2013)), (Yu, Bi, and Tresp (2006)), and (Tomanek and Hahn (2009)).

Another related research topic is developing a framework for active learning in domain adaption. Active domain adaptation was studied in (Zhang (2010)), (Rai et al. (2010)) and (Saha et al. (2011)). (Zhang (2010)) learned multiple related tasks simultaneously. (Saha et al. (2011)) and (Rai et al. (2010)) studied online active domain adaptation. In addition, (Saha et al. (2011)) proposed to use a free oracle to answer the target domain queries by taking advantage of source domain classifier.

In this paper, we add a projection step before active learning. A motivation behind this step is the prior belief that the model ought to be accurate by using global information.

## Problem Definition

In this section, we present required definitions and formulate the problem of multi-metric learning for cross-domain recommendation. Without confusion, we assume there are two domains, the source domain and the target domain. Our goal is to predict how people will act in the target domain and then discover potential users for the new product (i.e., target domain) that they never use.




(a) The neighborhood of $\mathbf{x}_i^o$ with Euclidean distance

(b) The neighborhood of $\mathbf{x}_i^o$ with Mahalanobis distance

Figure 1: **The Homogeneous Neighborhood of $\mathbf{x}_i^o$ with Euclidean and learned Mahalanobis distance. The squares with red and blue correspond to the impostors. The blob with yellow in the middle of the circle is $\mathbf{x}_i^o$. Our goal is to learn local distance metrics $\mathbf{A}_i$ which pulls the yellow blobs towards $\mathbf{x}_i^o$ while pushes the red and blue squares away from $\mathbf{x}_i^o$ in $\mathcal{N}_i^o$**

**Definition 1. Source / Target domain.** *A source (or target) domain can be denoted as* $D^S = \{(\mathbf{x}_1^S, y_1^S), \ldots, (\mathbf{x}_n^S, y_n^S)\}$ *which is a set of n data instances.* $\mathbf{x}_i^S \in \mathcal{X}^S$ *is a vector of attribute value of an instance and* $y_i^S \in \mathcal{Y}^S$ *is the corresponding real-value score. We denote* $(\mathbf{x}_j, y_j) = \{(\mathbf{x}_j^S, y_j^S), (\mathbf{x}_j^T, y_j^T)\}$ *as* $j^{th}$ *instance in source domain and target domain.*

The two superscripts $^S$ and $^T$ are used to differentiate the source and target domain, if there is no ambiguity. Suppose there are several patterns (labels) of data instances (users in this paper) which can be learned from the feature dimensions, our algorithm aims to learn local distance metrics to enhance the discriminability for each pattern. To define such local discriminability, we have the following definition:

**Definition 2. Homogeneous Neighborhood.** *Suppose* $\mathbf{x}_i^o$ *is a representative point. The homogeneous neighborhood of* $\mathbf{x}_i^o$*, denoted as* $\mathcal{N}_i^o$*, is the* $|\mathcal{N}_i^o|$ *data points similar with* $\mathbf{x}_i^o$*.* $|\mathcal{N}_i^o|$ *is the size of* $\mathcal{N}_i^o$*. With the local distance metric* $\mathbf{A}_t$*, the absolute deviation between* $\mathbf{x}_i^o$ *and the data points in the homogeneous neighborhood shall not exceed a threshold* $\varphi_i$*.*

For brevity, The homogeneous neighborhood can be taken as a circle where $\mathbf{x}_i^o$ is the center and $\varphi_i$ is the radius. Based

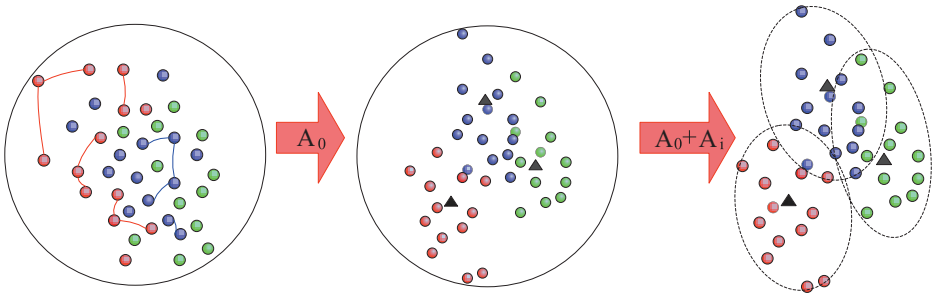



Figure 2: **The basic algorithm flowchart. The matrix $\mathbf{A}_0$ is the global distance metric capturing the communality among the neighborhoods, whereas $\mathbf{A}_t$ for $t > 0$ adds the local specific distance transformation. To utilize global information, we take $\mathbf{A}_0 + \mathbf{A}_t$ as parameters in local distance metric learning.**

on the above definition, we can define the local compactness of point $\mathbf{x}_i^o$ as:

$$\mathcal{C}_i = \{\mathbf{x}_j | d_{\mathbf{A}_t}(\mathbf{x}_i^o, \mathbf{x_j}) < \varphi_i\} \tag{1}$$

and formulate homogeneous neighborhood of point $\mathbf{x}_i^o$ as :

$$\mathcal{N}_i^o = \{\mathbf{x}_j | d_{\mathbf{A}_t}(\mathbf{x}_i^o, \mathbf{x}_j) < \varphi_i \ , |y_i^o - y_j| < \delta_i\} \tag{2}$$

The goal of our algorithm is to classify users into different homogeneous neighborhoods and learn specific distance metrics for each neighborhood. It is equivalent to minimize the local compactness within homogeneous neighborhood and maximize the scatterness among each neighborhood simultaneously. These metrics learned should rescale directions to push impostors further away than target neighbors by a large margin. Figure 1 provides an intuitive graphical illustration of the theme behind our algorithm. In this way, the potential users are ranked and recommended according to their location in the neighborhoods.

## Metric + Active Learning

**Algorithm Overview**. The basic procedure of our algorithm is to iterate the following procedure:

- We first define the similarity for two data points $(\mathbf{x}_i^S, y_i^S)$, $(\mathbf{x}_i^T, y_i^T)$ and $(\mathbf{x}_j^S, y_j^S)$, $(\mathbf{x}_j^T, y_j^T)$ to form an equivalence or inequivalence constrain by their labels.

$$(\mathbf{x}_i, \mathbf{x}_j) \in \begin{cases} \mathbf{S} & (y_i^S - y_j^S) + (y_i^T - y_j^T) < \delta \\ \mathbf{D} & (y_i^S - y_j^S) + (y_i^T - y_j^T) > \delta \end{cases} \tag{3}$$

  In the above, **S** and **D** denote the sets of similar and dissimilar constraints. Then a global distance metric $\mathbf{A_0}$ can be learned.
- Select $t$ representative points by supervised active learning. For each selected point, we partition the data points into neighborhoods $\{\mathcal{N}_i^o\}_{i=1...t}$ where the selected points $\{\mathbf{x}_i^o\}_{i=1...t}$ are the centers. Then radius of each neighborhood can be defined as $\varphi_i = \max\{d(\mathbf{x}_i^o, \mathbf{x}_j) | \mathbf{x}_j \in \mathcal{N}_i^o\}$
- Learn multiple distance metrics $\{\mathbf{A}_i\}_{i=1,...,t}$ for each local neighborhood while increase the radius to cover more samples.

Figure 2 shows the graphical view of the basic algorithm flowchart. Notice that each data point represents a user and the color is her label (e.g., according to user behaviors). In this following, we will introduce each step in detail.

### Global Distance Metric

In this subsection we learn a global distance metric that respects these constrains of similar pairs of points in $\mathbb{R}^n$. Xing et al. (2002) formulates distance metric learning into a constrained convex programming problem. Here, we define the distance in $\mathcal{N}_0^o$ as

$$d_{\mathbf{A}_0}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)\mathbf{A}_0(\mathbf{x}_i - \mathbf{x}_j)} \tag{4}$$

where $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{N}_0^o$ and $\mathbf{A}_0$ is the distance metric in $\mathcal{N}_0^o$. A simple way of defining a criterion for the desired metric is to demand that pairs of points $(\mathbf{x_i}, \mathbf{x_j}) \in S$ have small squared distance between them: $minimize_{\mathbf{A}} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in S} d_{\mathbf{A}}(\mathbf{x}_i, \mathbf{x}_j)$.

The global distance metric is used for capturing the commonality among all the users. In each iteration of our algorithm, it is also used to enhance the learning process of each specific local distance metric.

### Supervised Active Learning

In this subsection, we combine distance metric learning with the active learning paradigm. Our goal is to learn local metrics for each of the neighborhoods in order to minimize the distance between the representative point and its similar points. For selecting representative points, we mainly use an active learning method called *Transductive Experimental Design (TED)* (Yu, Bi, and Tresp (2006)), which aims to select the examples that are most uncertain to classify. Despite the empirical success, TED still has some limitation:

- In fact, TED just uses the global date information and tries to select $k$ representative date examples (which make the linear reconstruction loss minimized). In that sense, TED is a unsupervised algorithm.
- Without the label information provided, the date points selected trend to be at the border of the distribution.

Based on these analysis, we propose another type of TED. Suppose $\mathbf{A}_0$ is the global distance metric, we add a projection step before the usage of TED which is a linear transformation $\mathbf{x} \rightarrow \mathbf{A}_0^{1/2}\mathbf{x}$.

### Multi-Metric Learning

In this subsection we first formalize the problem of learning a specific distance metric for each neighborhood (user group) by an optimization problem. Then, we show to way to get the top-$k$ potential users.

$\min_{\mathbf{A_1},...,\mathbf{A_t}} \sum_{i=1}^{t} [\gamma_i \|\mathbf{A}_0 + \mathbf{A}_i\|_F^2 + \sum_{\mathbf{x_j} \in \mathcal{N}_i^o} d^2_{\mathbf{A}_i}(\mathbf{x}_i^o, \mathbf{x}_j) - u_i \sum_{\mathbf{x}_{j'} \notin \mathcal{N}_i^o} d^2_{\mathbf{A}_i}(\mathbf{x}_i^o, \mathbf{x}_{j'})]$

**subject to:**

**(1)** $\sum_{\mathbf{x}_j \in \mathcal{N}_i} d^2_{\mathbf{A}_i}(\mathbf{x}_i^o, \mathbf{x_j}) \geq 1$

**(2)** $u_{i=1,...,t} \geq 0$

**(3)** $\mathbf{A_1}, \mathbf{A_2} \cdots, \mathbf{A_t} \geq 0$

Table 1: **Convex optimization problem of CML.**

An important aspect of multi-domain metric learning is the appropriate coupling of the multiple domain data sets. To this end, we have to ensure that the learning algorithm does not put too much emphasis onto individual parameters $\mathbf{A}_1, \ldots, \mathbf{A}_t$. To ensure this balance, we use the regularization term and add the global metric $\mathbf{A}_0 \succ 0$ with each parameter as stated below:

$$\min_{\mathbf{A}_1,...,\mathbf{A}_t} \sum_{i=1}^{t} \gamma_i \|\mathbf{A}_0 + \mathbf{A}_i\|_F^2 \tag{5}$$

The trade-off parameter $\gamma_t$ controls the regularization of $\mathbf{A}_i$ for all $i = 0, 1, ..., t$. In practice, $\gamma_i$ is dependent on the size of the $i^{th}$ neighborhood $|\mathcal{N}_i^o|$. If $\gamma_i \to \infty$, the local-specific metrics $\mathbf{A}_i \succ 0$ become irrelevant zero matrices. Therefore if $\|\mathbf{A}_i\|_F \to \infty$, we learn a single $\mathbf{A}_i$ across all domains.

**Theorem 1.** *Given t representative points $\mathbf{x}_t$ and distance functions $d_{A_t}(\cdot,\cdot)$ as defined in Eq 4, the problem can be formulated as multi-metric learning.*

*Proof.* The optimal metrics learned depend on their representative points and it returns a radius for each neighborhood. Without confusion, we use the same notation $\mathbf{x}_i^o$ to denote the representative sample. Given any distance metrics, a circle can be made with $\mathbf{x}_i^o$ as the center. And the radius can be evaluated with the point locating farthest from $\mathbf{x}_i^o$, which suggests that $\varphi_i = \max\{d(\mathbf{x}_i^o, \mathbf{x}_j) | \mathbf{x}_j \in \mathcal{N}_i^o\}$. Furthermore, any positive semi-definite matrix $\mathbf{A}$ can be decomposed as $\mathbf{A} = \mathbf{L}^\top \mathbf{L}$. Therefore, it follows that there exists some matrix $\mathbf{L}_i$ such that $\mathbf{L}_i^\top \mathbf{L}_i = \mathbf{A}_0 + \mathbf{A}_i$. Hence we can push (or pull) any data instance with respect to $\{\mathbf{x}_i^o\}_{i=1,...,t}$ while learning distance metrics. Essentially, we can compute the linear projection $\mathbf{x}_i \to \mathbf{L}_t \mathbf{L}_{t-1} \ldots \mathbf{L}_1 \mathbf{x}_i$ iteratively. □

**Loss Function.** The loss function consist two terms, one acts to pull the similar neighbors to the center, and the other acts to push the dissimilar examples away from the center. We discuss them in turn.

The first term in loss function penalizes the distance between the input and its target. With the distance metric $\mathbf{A}_i$ given, the distance between the two samples is:

$$\varepsilon_{\mathbf{pull}} = \sum_{\mathbf{x}_j \in \mathcal{N}_i^o} \|\mathbf{A}_i(\mathbf{x}_i^o \mathbf{x}_j)\|^2 \tag{6}$$

The gradient of $\varepsilon_{\mathbf{pull}}$ is a pulling force that attract the similar neighbors in the transformed space with the metric distance $\mathbf{A}_0 + \mathbf{A}_i$. In practice, only the distance between each input and the center is penalized rather than the distance between the inputs, because accurate projection dose not require that all the samples in neighborhood should be similar.

The second term in loss function penalizes the distance between dissimilar samples in neighborhood. We put more pushing force on the imposter that is closer than the similar ones. Suppose $\mathbf{x}_j, \mathbf{x}_l$ are projected into $\mathcal{N}_i^o$ where $\mathbf{x}_j$ is similar with $\mathbf{x}_i^o$ and $\mathbf{x}_l$ is a imposter. We penalize the distance $d_{\mathbf{A}_i}(\mathbf{x}_i^o, \mathbf{x}_l)$, if $\mathbf{x}_l$ is closer to $\mathbf{x}_i^o$ than $\mathbf{x}_j$. In terms of this notation, the second term $\varepsilon_{\mathbf{push}}$ is given by:

$$\varepsilon_{\mathbf{push}} = \sum_{\mathbf{x}_j, \mathbf{x}_l \in \mathcal{N}_i^o} \sum_{\mathbf{x}_l \rightsquigarrow \mathbf{x}_i^o} [1 + \|\mathbf{L}(\mathbf{x}_i^o - \mathbf{x}_j)\|^2 - \|\mathbf{L}(\mathbf{x}_i^o - \mathbf{x}_l)\|^2]_+ \tag{7}$$

Where $\mathbf{x}_l \rightsquigarrow \mathbf{x}_i^o$ indicate that the imposter $\mathbf{x}_l$ is closer to $\mathbf{x}_i^o$ than some other similar samples. $[z]_+ = max(z, 0)$ denotes the hinge loss which monitors the equality in Eq 7, then the hinge loss is negative and makes no contribution to the overall loss function. The sub-gradient of Eq 7 generates a pushing force that repels the dissimilar sample out of the neighborhood.

We combine the above two terms in a single loss function for multi-metric learning. A weighting parameter $\mu \in (0, 1)$ is used to balance the goal:

$$\varepsilon(\mathbf{L}) = (1 - \mu)\varepsilon_{\mathbf{pull}}(\mathbf{L}) + \mu\varepsilon_{\mathbf{push}}(\mathbf{L}) \tag{8}$$

**Optimization Problem.** We combine the regularizer in Eq 5 and loss function in Eq 8 with the objective. We refer to the final optimization algorithm as Cross-domain Multi-metric Learning(CML) which is shown in Table 1.

**Theorem 2.** *the CML optimization problem is convex.*

*Proof.* The proof of Theorem 2 is completed in two steps. First, constraints of type (2) and (3) are standard linear and positive-semidefinite constraints, which are known to be convex. Convexity remains to be shown for constraints of type (1) and the objective. Both access the matrices $A_t$ exclusively in terms of the squared distance $d^2()$. This can be expressed as

$$d_t^2(\mathbf{x}_i, \mathbf{x}_j) = trace(\mathbf{A}_0 \mathbf{v}_{ij} \mathbf{v}_{ij}^\top) + trace(\mathbf{A}_0 \mathbf{v}_{ij} \mathbf{v}_{ij}^\top) \tag{9}$$

where $\mathbf{v}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)$ is linear in terms of the matrices $\mathbf{A}_t$ and it follows that the constraints of type (1) are also linear and therefore trivially convex. Similarly, it follows that all terms in the objective are also linear with the exception of the Frobenius norms in the regularization term. The sum of convex functions is convex, hence this concludes the proof. □

**User Recommendation.** Our final goal is to select top-$k$ target customers by predicting users' preference (label) from previous records. In the following, we show how to exploit the local metrics of each homogeneous neighborhood that

are learned by the above iterative algorithm for user recommendation. Without confusing, suppose there is only one source domain and the local distance metric learned for the homogeneous neighborhood of $\mathbf{x}$ can be stated as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{S,S} & \mathbf{A}^{S,T} \\ \mathbf{A}^{T,S} & \mathbf{A}^{T,T} \end{bmatrix} \tag{10}$$

Where $\mathbf{A}^{S,S}$ and $\mathbf{A}^{T,T}$ are distance metrics learned from the source domain data and the target domain data, respectively. Given a new data sample $\mathbf{x}^S$ with only records in source domain, it can be mapped in $k$ different neighborhoods denoted as $\{\mathcal{N}_i^o\}_{i=1,\ldots,m} = \{\mathcal{N}_i^o | d_{\mathbf{A}_i^{S,S}}(\mathbf{x}^S, \mathbf{x}_i^o) \leq \varphi_i\}$. The predicting score of $\mathbf{x}^S$ in target domain depends on the location in its homogeneous neighborhoods. In this way, we only need to solve the deviation between $\mathbf{x}$ and $\mathbf{x}_i^o$ where $i = 1, \ldots, m$. Then, the expected value of $\mathbf{x}$ in target domain can be evaluated with:

$$y^T = \frac{1}{m} \sum_{\mathbf{x} \in \mathbf{N}_i^o}^{m} \frac{d_{\mathbf{A}^{S,S}}(\mathbf{x}^S, \mathbf{x}_i^o)}{\varphi_t} \cdot y^S \tag{11}$$

where $(\mathbf{x}^S, y^S)$ is the data sample (predicted user)'s records in source domain. Ranking users based on the predicted labels (i.e., $y^T$), we could get the recommendation results.

## Exprimental Results

**Experimental Datasets.** We select a three products (web applications, namely INPUT, VIAFLY and CMCC) dataset provided by a famous speech recognition company. VIAFLY is a newly launched application (i.e., target domain) while INPUT and CMCC have many loyal customers (i.e., source domain). Although these applications share some certain functions, there still exist individual features. The aim is to promote VIAFLY by selecting top-$k$ potential customers who have used INPUT and CMCC. In the experiment, we focus on customers who have used all these applications for dealing with the user skewness. Thus, we predict the user preferences to VIAFLY. After eliminating the duplicate and null records, there are 7,252 users' records remained. In addition, we use the users' interaction frequency in different time schedule (e.g., afternoon and middle-night) as features, which result in a 122 dimensional space. To characterize the individual user's feature, we made the first download period (the time that given application is downloaded) as the centering axis for this user.

**Baselines.** In the following, we call our multi-metric learning algorithm for cross-domain recommendation as CML, and we choose the following four baselines:

- **Content Similarity with Metric Embedding (Content+ME):** Notice that there's no usage of label information in Content-based methods. To improve the performance of these methods, we embed the Mahalanobis distance metric in them. The basic idea is that the accuracy increases by taking more advantage of similar data instances with test samples. Similarity score is the Cosine similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ stated as $Sim(\mathbf{i}, \mathbf{j}) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{||\mathbf{x}_i|| ||\mathbf{x}_j||}$.
- **Collaborative Filtering with Metric Embedding (CF+ME):** It leverages the existing several domains to make the recommendation. We also employ the metric learned above for the CF method.
- **Boosting for Transfer Learning (TrAdaBoost):** TrAdaBoost attempts to iteratively reweight the instances in source domain to reduce the effect of the "bad" source instances while encourage the "good" source instances to contribute more. (Eaton and desJardins (2011))
- **Weighted Adjusted KNN (WAkNN):** WAkNN is on the k-NN classification paradigm (Zuo et al. (2013)). In WAkNN, the importance for each of the point in classification of a training cluster can be learned and the weight vector respecting the importance is maintained. The weight vector is evaluated in iteration so that important points contribute more in the similarity measure.

**Evaluation Metrics.** We randomly select 10% users to be the test set and the remaining 90% users for training. Meanwhile, we treat the entire record duration, which is actually half a year, as one month (e.g., 30 days). Thus, we use the number of days (from 0 to 30) that each user opens the corresponding application as label (e.g., $y^S$ or $y^T$). Compared with the simple label prediction accuracy, product providers usually care more about two abilities of a recommendation model: Can we extract the customers that are really interested with the new product and what's the percentage of one extraction? To evaluate such local permutations, we first introduce the definition of a novel curve:

**Definition 3. top-$k$ Average Prediction Curve (top-$k$ APC).** *We are given a set of items $\mathcal{X}$. Let $\mathbf{X}_k$ be top-k candidates which is a finite subset of $\mathcal{X}$. Suppose $\mathcal{Y}^{\mathrm{T}}$ and $\widehat{\mathcal{Y}}$ are the true and predicted score of $\mathcal{X}$ respectively. Then $\mathcal{X}$ is permuted via $\widehat{\mathcal{Y}}$ and $\mathbf{X}_k$ can be drawed whose true score is denoted as $\mathbf{Y}_k^T$. The top-k APC is plotted to show how the average value of $\mathbf{Y}_k^T$ varies with k increasing.*

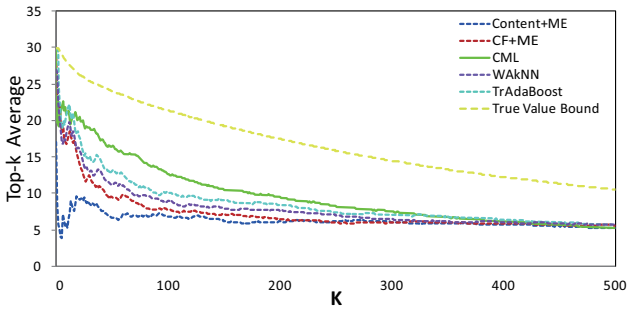



Figure 3: **Comparisons of the top-$k$ APC with $k$ increasing.**

Similar with NDCG, top-$k$ APC pays more attention to the first several candidates in the ranked list. However top-$k$ APC only cares the average value of $\mathbf{X}_k$, which means the order of the items in $\mathbf{X}_k$ is not considered. Moreover, top-$k$ APC is used for target customer orientating which is sensitive with respect to the cost of marketing. Besides providing

| Neighborhood Number | Content+ME | CF+ME | TrAdaBoost | Weighted-KNN | CML |
|---|---|---|---|---|---|
| 10 | $0.1900 \pm 0.0004$ | $0.1846 \pm 0.0001$ | $0.1812 \pm 0.0002$ | $0.2103 \pm 0.0004$ | $\mathbf{0.180 \pm 0.0002}$ |
| 30 | $0.1804 \pm 0.0017$ | $0.1784 \pm 0.0002$ | $0.1876 \pm 0.0003$ | $0.1837 \pm 0.0001$ | $\mathbf{0.1656 \pm 0.0002}$ |
| 50 | $0.1820 \pm 0.0003$ | $0.1778 \pm 0.0002$ | $0.1643 \pm 0.0001$ | $0.1446 \pm 0.0002$ | $\mathbf{0.1246 \pm 0.0002}$ |

Table 2: **Overall RMSE results.**

performance of prediction, top-$k$ APC can also assist to decide what the percentage of potential users can be extracted as the target customers. Thus it is related in a direct and natural way to cost/benefit analysis of diagnostic decision making.

We also adopt RMSE as our evaluation metric to measure the prediction accuracy. Particularly, Root Mean Square Error (RMSE) is defined as:

$$\textbf{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(\mathbf{y}_i^T - \widehat{\mathbf{y}}_i)^2}{|\mathcal{L}|}} \tag{12}$$

where $\mathbf{y}_i^T$ and $\widehat{\mathbf{y}}_i$ are the true and predicted score respectively, and $\frac{1}{|\mathcal{L}|}$ is a normalization factor. The smaller is the value, the better is the performance.

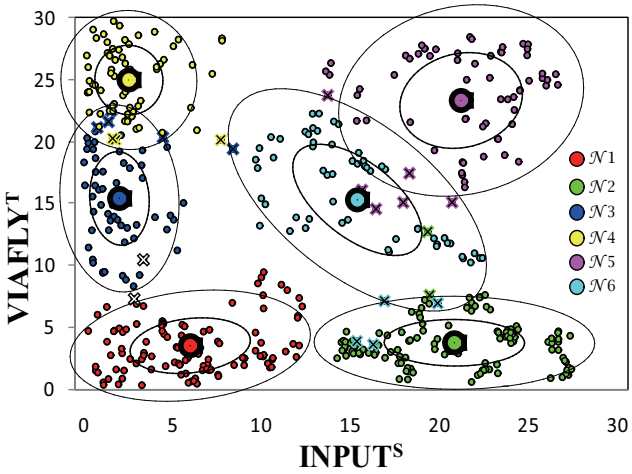



Figure 4: **CML on the real world data set. With n = 500 data points sampled from the distribution. Multiple local distance metrics learned from different neighborhoods which are selected after eliminating a few minor ones. The figure shows the homogeneous neighborhoods extracted by our method for distance metric learning.**

## Recommendation Performance Analysis

- **Experiments on samples.** For visualization, we randomly select 500 data samples from the data set. Meanwhile, we only take INPUT application as the source domain and VIAFLY as the target domain for brevity. To illustrate the sparse connection across domains by user behavior distribution, we denote $INPUT^S$ and $VIAFLY^T$ be the axes and the results are shown in Figure 4. From this figure, we can see that only a few customers frequently use both of these two applications. Furthermore, we label the data points with different colors after using the active learning method. The selected representative data points can be visualized by the circles in Figure 4, and the ellipsoids show the effect of the distance metrics learned by CML. The crosses are the imposters respecting to the representative points in the centers. Notice that there may exist points that are not in any neighborhoods, however, CML is still able to predict the labels of them in the target domain.

- **Cross-domain neighborhoods analysis.** We try to answer the question of how many neighborhoods are enough for the cross-domain recommendation. To this end, we perform an analysis by varying the number of homogeneous neighborhoods in the proposed CML method. Table 2 shows its RMSE performance with the number of cross-domain neighborhoods varied. As we see, performance of CML is close to TrAdaBoost when the number is small (i.e., 10) and increasing this number can improve the performance of CML's recommendation results. Actually, it tends to be stable after the number is near to and larger than 50. This also demonstrate the stability of CML method in terms of the number of neighborhoods.

- **Overall comparison.** Table 2 and Figure 3 show the performance comparisons between CML with the baselines on RMSE and top-$k$ APC, respectively. Have said that when the number of neighborhoods are comparatively large, the CML could clearly outperforms the other baseline methods with 2%-6% of improvement (in terms of RMSE). This improvement is more obvious in Figure 3 with the top-$k$ APC of CML much higher than the baselines. In summary, Content-based methods only consider content information leading to the worst performance, and CML differentiates "Homogeneous Neighborhoods" from those irrelevant neighborhoods and thus obtains significant improvement over two related baselines, i.e., TrAdaboost and WAkNN.

## Conclusion

In this paper, we studied the problem of cross-domain recommendation by multi-metric learning. For better identifying the top-$k$ potential users, our proposed solution combines metric learning and active learning. Specifically, we first learnt a global distance metric for capturing the similarity among all the users. Then, a supervised active learning is adopted to select representative users and the local distance metrics are thus got in each user group. In this way, the potential users are ranked based on the predicting scores in the target domain. Finally, the experimental results on the real world datasets validate the effectiveness of our method.

## References


Bhagat, S.; Weinsberg, U.; Ioannidis, S.; and Taft, N. 2013. Recommending with an agenda: Active learning of private attributes using matrix factorization. *arXiv preprint arXiv:1311.6802*.

Blitzer, J.; Weinberger, K. Q.; and Saul, L. K. 2005. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, 1473–1480.

Der, M., and Saul, L. K. 2012. Latent coincidence analysis: A hidden variable model for distance metric learning. In *Advances in Neural Information Processing Systems*, 3239–3247.

Eaton, E., and desJardins, M. 2011. Selective transfer between learning tasks using task-based boosting. In *AAAI*.

Hong, Y.; Li, Q.; Jiang, J.; and Tu, Z. 2011. Learning a mixture of sparse distance metrics for classification and dimensionality reduction. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, 906–913. IEEE.

Lajugie, R.; Arlot, S.; and Bach, F. 2013. Large-margin metric learning for partitioning problems. *arXiv preprint arXiv:1303.1280*.

Law, M. T.; Gutierrez, C. S.; Thome, N.; and Gancarski, S. 2012. Structural and visual similarity learning for web page archiving. In *Content-Based Multimedia Indexing (CBMI), 2012 10th International Workshop on*, 1–6. IEEE.

McFee, B.; Barrington, L.; and Lanckriet, G. 2012. Learning content similarity for music recommendation. *Audio, Speech, and Language Processing, IEEE Transactions on* 20(8):2207–2218.

Nguyen, N., and Guo, Y. 2008. Metric learning: A support vector approach. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 125–136.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on* 22(10):1345–1359.

Park, K.; Shen, C.; Hao, Z.; and Kim, J. 2011. Efficiently learning a distance metric for large margin nearest neighbor classification. In *AAAI*.

Rai, P.; Saha, A.; Daumé III, H.; and Venkatasubramanian, S. 2010. Domain adaptation meets active learning. In *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, 27–32. Association for Computational Linguistics.

Saha, A.; Rai, P.; Daumé III, H.; Venkatasubramanian, S.; and DuVall, S. L. 2011. Active supervised domain adaptation. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 97–112.

Tang, J.; Wu, S.; Sun, J.; and Su, H. 2012. Cross-domain collaboration recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 1285–1293. ACM.

Tarlow, D.; Sutskever, I.; and Zemel, R. S. 2013. Stochastic k-neighborhood selection for supervised and unsupervised learning.

Tomanek, K., and Hahn, U. 2009. Semi-supervised active learning for sequence labeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, 1039–1047. Association for Computational Linguistics.

Wang, F.; Sun, J.; Li, T.; and Anerousis, N. 2009. Two heads better than one: Metric+ active learning and its applications for it service classification. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, 1022–1027. IEEE.

Weinberger, K. Q., and Saul, L. K. 2009. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research* 10:207–244.

Xing, E. P.; Jordan, M. I.; Russell, S.; and Ng, A. 2002. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, 505–512.

Yu, K.; Bi, J.; and Tresp, V. 2006. Active learning via transductive experimental design. In *Proceedings of the 23rd international conference on Machine learning*, 1081–1088. ACM.

Zhang, Y. 2010. Multi-task active learning with output constraints. In *AAAI*.

Zuo, X.; Feng, B.; Yao, Y.; Zhang, T.; Zhang, Q.; Wang, M.; and Zuo, W. 2013. A weighted ml-knn model for predicting users personality traits. In *2013 International Conference on Information Science and Computer Applications (ISCA 2013)*. Atlantis Press.