Latent Space Domain Transfer between High Dimensional Overlapping Distributions

Sihong Xie[†] Wei Fan[‡] Jing Peng[*] Olivier Verscheure[‡] Jiangtao Ren[†][*]
[†]Sun Yat-Sen University, Guangzhou, China
[‡]IBM T. J. Watson Research Center, New York, USA
[*]Montclair State University, Montclair, New Jersey, USA
{mc04xsh@mail2, issrjt@mail}.sysu.edu.cn
{weifan,ov1}@us.ibm.com,pengj@mail.montclair.edu

ABSTRACT

Transferring knowledge from one domain to another is challenging due to a number of reasons. Since both conditional and marginal distribution of the training data and test data are non-identical, model trained in one domain, when directly applied to a different domain, is usually low in accuracy. For many applications with large feature sets, such as text document, sequence data, medical data, image data of different resolutions, etc. two domains usually do not contain exactly the same features, thus introducing large numbers of "missing values" when considered over the union of features from both domains. In other words, its marginal distributions are at most overlapping. In the same time, these problems are usually high dimensional, such as, several thousands of features. Thus, the combination of high dimensionality and missing values make the relationship in conditional probabilities between two domains hard to measure and model. To address these challenges, we propose a framework that first brings the marginal distributions of two domains closer by "filling up" those missing values of disjoint features. Afterwards, it looks for those comparable sub-structures in the "latent-space" as mapped from the expanded feature vector, where both marginal and conditional distribution are similar. With these sub-structures in latent space, the proposed approach then find common concepts that are transferable across domains with high probability. During prediction, unlabeled instances are treated as "queries", the mostly related labeled instances from out-domain are retrieved, and the classification is made by weighted voting using retrieved out-domain examples. We formally show that importing feature values across domains and latent-semantic index can jointly make the distributions of two related domains easier to measure than in original feature space, the nearest neighbor method employed to retrieve related out domain examples is bounded in error when predicting in-domain examples. Software and datasets are available for download.

Categories and Subject Descriptors

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

General Terms

Algorithms

1. INTRODUCTION

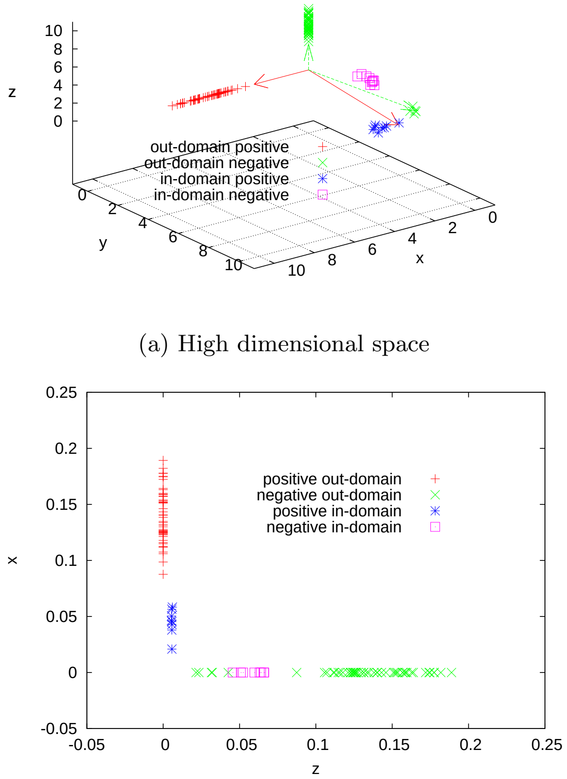
Domain transfer for high dimensional datasets, such as microarray data, text data, web log data, is a challenging problem. When the number of features is in thousands, in-domain and out-domain data rarely share the exact set of feature. In the same time, there may not be any labeled example from in-domain. When one uses the union of the overlapping and non-overlapping features and leave the missing values as "zero", the distance of two marginal distributions $p(\mathbf{x})$ can become asymptotically very large. Otherwise, if one only considers those common features, such information may be limited in its predictability, and many algorithms may have difficulties to find transferable structures across the two domains. Therefore, one main challenge to transfer high dimensional overlapping distribution is on how to effectively use those large number of features that are non-overlapping or present in one domain but not in the other. Nonetheless, the main task for inductive learning is to identify "sub-structures" between the two domains where it is transferable or the conditional probabilities $p(y|\mathbf{x})$ across these structures are similar. This is particularly difficult under the given scenario. First, there are no labeled data from in-domain, hence the relationship of $p(y|\mathbf{x})$ across the two domains are not directly measurable. Second, the problems are high dimensional with missing values. Thus the second main challenge is on how to look for transferable substructures (or considering subset of features in conditional probability) in this space.

In order to resolve these two main challenges, we first bring the marginal distribution of the overlapping distributions asymptotically closer by "filling up" the missing values in some reasonable way to be discussed. Then to resolve the high dimensional problem, we map the original feature space into a low-dimensional "latent space" where each feature is a linear combination of high dimensional features. We show that in this low dimensional space, transferable concepts are easier to discover and their prediction error can be bounded. By default, most inductive learner usually makes

[*]The author is supported by the National Natural Science Foundation of China under Grant No. 60703110

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

the cluster assumption indicating that two nearby points are likely to have the same class label. This is more likely to hold true in low-dimensional space. Specifically, it has been shown formally that as the dimensionality of the space increases, the Euclidean distance between any points in the high dimensional space is getting asymptotically closer. In other words, distance-based classification is unreliable in high dimensional space. For example, in Figure 1(a), we plot two domains' data in 3-D space, where the pluses$(+)$ and crosses$(\times)$ are labeled out-domain positive and negative instances, respectively. The stars$(\star)$ and squares$(\Box)$ are the unlabeled in-domain positive and negative instances, respectively. The two domains' data are related since the positives from both domains lie on the x-y plane while all the negatives lie on y-z plane. However, two domains are different since they have quite different distribution. When classifying the stars using cosine similarity, because they are closer to some of the crosses than to the pluses, they are classified incorrectly, The cluster assumption is obviously violated in this space. Next, we will briefly summarize the proposed approach and come back to this example to see how the problem is being resolved.



(a) High dimensional space

(b) Low dimensional space

Figure 1: Illustrating Example

Given these challenges, we propose a new latent space based method for domain transfer where there is no labeled in-domain data and both out-domain and in-domain are high-dimensional. Briefly, the proposed approach has three main steps, as depicted in Figure 2. We first employ a "multiple regression method" to fill up those "missing values" across the two domains in order to draw the marginal distributions closer, here we assume that the discrepency between distributions of two domains over these overlapping features are reasonably small (these features are shared by two domains). Then, both the high dimensional out-domain and in-domain data are mapped into a low-dimensional latent-space. In order to classify an unlabeled in-domain example, we retrieve the closest neighbors in the latent space to the in-domain example and use weighted voting as the predicted class label. To be exact, regression models are built using out-domain data, taking overlapping features as independent variables and non-overlapping features as dependent variables. Missing values in in-domain are filled up by these models. Intuitively, these uniform transferable models describe feature dependency in both domains, thus data will lie in the same space and marginal distribution $p(\mathbf{x})$ become closer (see Section 3.2). Second, we propose to use SVD (Singular Value Decomposition) for dimension reduction and similar structure discovery. SVD first maps the high dimensional out-domain and in-domain data to a latent space with lower dimension. In this space, data giving the same concept will lie nearby[5] (see Section 3.3), i.e. the closer two instances are, the more likely they are having the same label, thus $p(y|\mathbf{x})$ across domains will be similar within cluster where instances are coherently nearby. To exploit this similar conditional distribution, given an in-domain instance $\mathbf{x}_0$, those out-domain instances which are most close to $\mathbf{x}_0$ are retrieved and $\mathbf{x}_0$ is classified by similarity-weighted voting. Now, let's re-visit the previous example. We apply SVD on two domains' data (step 3 in Figure 2) and the lower dimensional representation in latent space are obtained, as plotted in Figure 1(b). For each in-domain point $\mathbf{x}$(star or square), we retrieve p nearest out-domain points (plus or cross) and classify $\mathbf{x}$ according to the labels of the retrieved points and the corresponding similarity (step 4 in Figure 2). Note that the cluster assumption holds in this space. We can see clearly that the stars/squares are now approximately in the same direction of the pluses/crosses, all retrieved out-domain points will be pluses/crosses.

Our contributions are as follow: (1) We propose a transfer learning framework which make joint distributions of two high dimensional domains with overlapping features easier to measure, and thus identify transferable concepts is straightforward. Multiple regression and SVD are used to resolve the difficulties to measure and identify structures with similar $p(\mathbf{x})$ and $p(y|\mathbf{x})$ respectively. The experiments results demonstrate the proposed framework outperforms traditional learning algorithms including SVM. (2) Formal

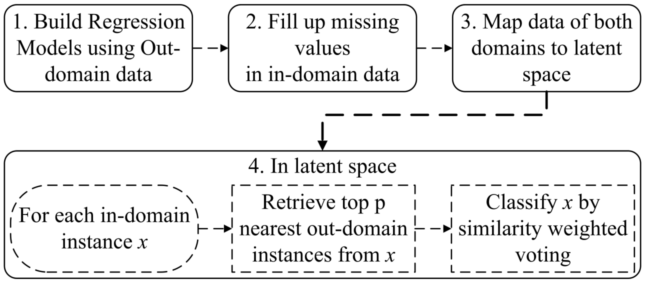

1. Build Regression Models using Out-domain data
2. Fill up missing values in in-domain data
3. Map data of both domains to latent space
4. In latent space
For each in-domain instance x
Retrieve top p nearest out-domain instances from x
Classify x by similarity weighted voting

Figure 2: Flow Chart of the Proposed Framework

Table 1: Symbols Definition

Symbol	Definition
(X^ℓ, Y^ℓ)	Labeled out-domain data
(X^u)	Unlabeled in-domain data
$(\hat{X}^u)$	X^u with predicted missing values
ℓ	Number of out-domain
u	Number of in-domain points
$p(\mathbf{x})$	marginal distribution
$p(y\|\mathbf{x})$	conditional distribution
F_+	Features exist only in out-domain
F_c	Features shared by two domains
W	input high dimensional space
S	low dimensional latent space
A	Data matrix combining X^ℓ and $\hat{X}^u$
U	Matrix of principle direction
Σ	Matrix of singular values
V	Matrix of principle component
k	Dimensionality of latent space

analysis demonstrates that missing values importation via regression asymptotically reduces the difference between two marginal distributions $p(\mathbf{x})$ than one without importation. In the low dimensional latent space, we give the upper bound of nearest neighbor classifier used in the given transfer learning scenario. This condition is guaranteed by the clusters recovery process[5]. (3) The proposed framework can be generalized to include various regression and cluster methods, not limited to the experimental choices.

2. LATENTMAP: MEASURE AND TRANSFER OVERLAPPING DISTRIBUTIONS

We introduce latent space based transfer learning framework between two domains that lie on two different spaces that are at most overlapping. We summarize symbols and their definitions in Table 1. Suppose we have ℓ labeled out-domain instances $(X^\ell, Y^\ell) = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\}$ and u unlabeled in-domain instances $X^u = \{\mathbf{x}_{\ell+1}, \ldots, \mathbf{x}_{\ell+u}\}$. Let X^ℓ and X^u also denote matrices, each row represents an instance while each column represents a feature. We assume that two domains' data fall in two categories ($y \in \{0, 1\}$). As an important step, we first discuss how to perform multiple regression to fill up the missing values, then SVD-based dimension-reduction and clustering is briefly discussed.

2.1 Missing Values Importation

Since two spaces are overlapping or they only share some small number of features, when a classification model is trained on out-domain data using all its features then subsequently use to prediction in-domain documents, one need to consider how to properly handle those "missing values" or features in F_+ that exist only in out-domain but not in in-domain. As we shall see in Section 3, if we leave those missing values as "zeros", the distance between any two points can be asymptotically very large and order of difference between different points are difficult to distinguish. Nonetheless, if we fill up the missing values in some reasonable way, the order of the difference between closer points and further points is more measurable. We propose to use "multiple regression" to fill up those missing values across the two domains. Let F_c denote the set of features overlap across domains, and H^ℓ and H^u denote the out-domain and in-domain data X^ℓ and X^u projected on F_c respectively. Let $F_+ = \{f_+^1, \ldots, f_+^{|F_+|}\}$. The values of out-domain points on F_+ can be seen as a series of column vector $y_+^l, l = 1, \ldots, |F_+|$ (not class label) while the values of in-domain points on F_+ are missing. $|F_+|$ regression models $M = \{M_1, \ldots, M_{|F_+|}\}$ can be built using H^ℓ as observation of independent variables (namely, features in F_c) and $y_+^l, l = 1, \ldots, |F_+|$ as observation of dependent variables (namely, features in F_+), thus M_l gives the estimated functional relationship between F_c and f_+^l. M_l is then applied on H^u to predict the values of all in-domain points on f_+^l, and thus missing values are filled up.

Various regression models can fit into the proposed framework, since they are basically seeking a function such that the predictions are expected to have deviation from the actual values within a given small tolerance ϵ for all the training data. For implementation, ϵ-Support Vector Regression (ϵ-SVR)[13, 14] is employed to fill up missing values, it can be formulated as:

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \sum_{i=1}^{n} (\xi_i + \xi_i^*) \\
\text{s.t.} \quad & y_+^{li} - \langle \mathbf{w}, \mathbf{h}_i \rangle - b \le \epsilon + \xi_i \\
& \langle \mathbf{w}, \mathbf{h}_i \rangle + b - y_+^{li} \le \epsilon + \xi_i^* \\
& \xi_i, \xi_i^* \ge 0
\end{aligned}
$$

where $\mathbf{h}_i$ is the i-th row in H^ℓ and y_+^{li} is the i-th entry of y_+^l. ξ_i, ξ_i^* are slack variables, $\mathbf{w}$ is the model parameter and λ determines the trade-off between the generalization ability (measure by $\|\mathbf{w}\|^2$) and the amount up to which deviations larger than ϵ are tolerated. The functional relationship learned using H^ℓ and y_+^l, i.e. M_l, is then applied on H^u (can be seen as new-coming examples in the same space with H^ℓ) and give prediction on y_+^l. The aim of multiple regression is to minimize the discrepency between the marginal distributions of two domains. Admittedly, H^ℓ and H^u come from different distributioins, however generalization error bound is given in [16] when training and test data are from different distribution, and this justifies our regression strategy.

2.2 Dimensionality Reduction

The marginal distributions $p(\mathbf{x})$ of two domains are made easier to measure and quantify in the input space W via missing values importation. However, the dimension of the vector space is still so high that makes it hard to identify similar structures across domains. Given a point $\mathbf{x} \in \mathbb{R}^n$, as n grows, the distance between the "nearest" neighbor to $\mathbf{x}$ will approximate to the distance from the "farthest" neighbor to $\mathbf{x}$. Thus, the distance in high-dimensional space is meaningless. Since for any reasonable problem, one makes the the clustering-manifold assumption that nearby points will have similar label, the Euclidean distance in the high dimensional space is no longer a good measure to even apply this assumption. In addition, there is no labeled in-domain data, no information of $p(y|\mathbf{x})$ is given, thus the relationship of $p(y|\mathbf{x})$ across domains are not directly measurable. To solve this problem, one ought to map the data to a low dimensional space. In this space, sub-structures are discovered in places where in-domain and out-domain points will have similar labels and the cluster assumption holds across true with high probability. We propose to use SVD for latent space mapping and sub-structures discovery.

In short, SVD first maps the data to a lower dimensional space, this mapping has been proved to be consistent with k-means clustering's objective function, namely, the low dimensional representation of data are cluster membership indicators from which clusters structure can be reconstructed [5]. By cluster assumption [1], points in the same cluster has similar conditional distributions regardless of whether the points are from the same domain or not, thus we have identify transferable sub-structure in latent space. Specifically, let $A \in \mathbb{R}^{t \times d}$ be the feature-instance matrix of two domains (see Table 1), then each row of A is an instance and each column of A is one dimension of the union of two space. We further assume that the the first ℓ rows are out-domain instances and the next u rows are in-domain instances and the first $|F_+|$ columns are features exist only in out-domain and the next $|F_c|$ columns represent those overlapping dimensions of two spaces, thus $d = \ell + u$ and $t = |F_+| + |F_c|$. SVD is applied on A

$$A = U \Sigma V^{\mathrm{T}} \tag{1}$$

where Σ is a $t \times d$ matrix with diagonal entries ($\sigma_i, i \in \{1, \dots, \min\{t, d\}\}$) being the singular values

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{t,d\}}$$

and off-diagonal entries being zero. The $t \times t$ matrix U and the $d \times d$ matrix V are the left and right eigenvector matrices, respectively. Note that both U and V are orthogonal matrices: $U^{\mathrm{T}}U = I_t$, $V^{\mathrm{T}}V = I_d$. Then SVD can be seen as a solution of PCA by the following equation:

$$A^{\mathrm{T}}A \ = \ (V\Sigma U^{\mathrm{T}}U)(\Sigma V^{\mathrm{T}}) = V\Sigma^2 V^{\mathrm{T}}$$

where $V = \{\mathbf{v}_1, \dots, \mathbf{v}_d\}$ is the eigenvectors. For dimension reduction, we obtain the top $k-1$ eigenvectors with largest eigenvalues $(\sigma_1^2, \dots, \sigma_{k-1}^2)$ as new data representation. That is let $V_{k-1} \in \mathbb{R}^{d \times k-1}$ contains $(\mathbf{v}_1, \dots, \mathbf{v}_{k-1})$ as columns, then each row $v_i, i = 1, \dots, d$ in V_{k-1} is a new representation of $\mathbf{x}_i, i = 1, \dots, d$ in the $k-1$-dimensional space, i.e.

$$V_{k-1} = A^{\mathrm{T}}U_{k-1}\Sigma_{k-1}^{-1} \tag{2}$$

where U_{k-1} is the first $k-1$ columns of U and Σ_{k-1} is a $k-1 \times k-1$ matrix keeping the first $k-1$ columns and rows of Σ. Note that we use $A^{\mathrm{T}}U_{k-1}\Sigma_{k-1}^{-1}$ to approximate the top $k-1$ columns of the principle component matrix instead of truncating the V to $k-1$ columns which equals $A_{k-1}^{\mathrm{T}}U_{k-1}\Sigma_{k-1}^{-1}$. This is common in information retrieval and justified by the fact that A_{k-1} is the best rank $k-1$ approximation of A. $\mathrm{U}_{k-1}\Sigma_{k-1}^{-1}$ can be considered as a mapping $T : W \rightarrow S$ which maps data in the space W (space where each dimension represents one feature) to a new lower dimensional space S (called latent space). For clustering, [5] has proved that the above dimension reduction also reveals the information of the clusters structure which k-means seeks to discover. Specifically, k-means uses k centroids to represent k clusters which are determined by minimizing the sum of squared error

$$J_k = \sum_{l=1}^{k} \sum_{i \in C_l} (\mathbf{x}_i - m_l)^2 \tag{3}$$

where $m_l = \sum_{i \in C_l} \mathbf{x}_i / n_l$ is the centroid of cluster C_l and n_l is the number of points in C_l. Let $Q_{k-1} = (\mathbf{q}_1, \dots, \mathbf{q}_{k-1})$ be cluster membership indicator vectors (we'll show how to reconstruct cluster structure later) such that $Q_{k-1}^T Q_{k-1} = I_{k-1} \in \mathbb{R}^{(k-1)\times(k-1)}$ and $\mathbf{q}_l^{\mathrm{T}}\mathbf{e} = 0, l = 1, \dots, k-1$. Then the k-means objective can be written as

$$J_k = \mathrm{Tr}(A^{\mathrm{T}}A) - \mathbf{e}^{\mathrm{T}}A^{\mathrm{T}}A\mathbf{e}/n - \mathrm{Tr}(Q_{k-1}^{\mathrm{T}}A^{\mathrm{T}}AQ_{k-1}) \tag{4}$$

since the first two terms in Equation (4) have nothing to do with Q_{k-1}, then the k-means objective becomes

$$\max_{Q_{k-1}} \mathrm{Tr}(Q_{k-1}^{\mathrm{T}}A^{\mathrm{T}}AQ_{k-1}) \tag{5}$$

The objective (5) has a closed form and global optimal solution which is the eigenvectors $V_{k-1} = (\mathbf{v}_1, \dots, \mathbf{v}_{k-1})$ of $A^{\mathrm{T}}A$. The clusters structure can be recovered by constructing a connectivity matrix:

$$Sim = V_{k-1}V_{k-1}^{\mathrm{T}} \tag{6}$$

The entry Sim_{ij} can be interpreted as connectivity between $\mathbf{x}_i$ and $\mathbf{x}_j$, we can associate a connectivity probability between $\mathbf{x}_i$ and $\mathbf{x}_j$: $c_{ij} = Sim_{ij}/Sim_{ii}^{\frac{1}{2}}Sim_{jj}^{\frac{1}{2}}$. Finally, the clusters structure is determined such that $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same cluster if and only if $c_{ij} > \beta$.

The effect of SVD is three-fold. First, further bring the marginal distributions $p(\mathbf{x})$ in S close given $p(\mathbf{x})$ is sufficient close in W, this is achieved by the mapping given by $U_{k-1}\Sigma_{k-1}^{\mathrm{T}}$ as we will see in Section 3. Second, identify transferable sub-structure through which out-domain knowledge can be used for in-domain learning. Given an unlabeled in-domain instance, we exploit the cluster structure by using top p nearest labeled instances for weighted voting.

$$Label'(\mathbf{x}_i) = \sum_{\mathbf{x}_j \in N(\mathbf{x}_i)} Label(\mathbf{x}_j) Sim(\mathbf{x}_i, \mathbf{x}_j) \tag{7}$$

where $Label'(\mathbf{x})$ is the prediction and $Label(\mathbf{x})$ is the true label of $\mathbf{x}$ and $N(\mathbf{x}_i)$ is the set of p nearest neighbors of $\mathbf{x}_i$ according to Sim. Since given $\mathbf{x}_i$, those $\mathbf{x}_j$ with highest c_{ij} must be in the same cluster of X_i, given the size of the cluster is large enough, thus points in $N(\mathbf{x}_i)$ must be those out-domain points which have the most similar conditional distribution with $\mathbf{x}_i$ by the cluster assumption[1]. Finally, we see that the dimension is simultaneously reduced to $k-1$.

2.3 LatentMap for Domain Transfer

The above discussion provides us some insights to solve the problems raised in Section 1. We propose to use both regression and latent space mapping to bring the distributions $p(\mathbf{x}, y)$ of in-domain and out-domain close and at the same time reduce the dimensions. We call the proposed framework LatentMap in the sequel. As shown in Algorithm 1, LatentMap consists of two key steps to deal with distributional difference between domains. First, X^{ℓ} is used to give models which describe the relationship between features F_c and each feature $f_+^l, l = 1, \dots, |F_+|$, then we apply these models on X^u to fill up the missing values, the resulting data is denoted by $\hat{X}^u$. This step ensure we can bring the in-domain and out-domain marginal distributions in W close, as shown in Section 3. Then these data is used to construct the data matrix A, which incorporates X^{ℓ} as its first ℓ rows and $\hat{X}^u$ as its next u rows. A is decomposed into three matrices using SVD and new representation of two domains' data are obtained in the latent space S. It has to be noted that this representation is indicators of clusters structure, instances having the same concept will be close in the latent space S. In other words, points in the same cluster express similar concepts and their labels are expected to be the same, thus we bring the conditional distributions $p(y|\mathbf{x})$ of two domains close in S. Note also that this new representation of data have much lower dimensions than the data have in W and dimension reduction is fulfilled. What follows is instance retrieval step, given an in-domain instance $\mathbf{x}_j, j = \ell+1, \ldots, \ell+u$, the labels of p out-domain instances which are most close to $\mathbf{x}_j$ are use to vote for the label of $\mathbf{x}_j$. This retrieval process may use instances from other clusters which $\mathbf{x}_j$ does not belong to according to the cluster recovery process, nonetheless, the way that points are clustered depends on β (see the previous section): if β is small, size of clusters will become large and otherwise small, the clustering result may not be perfect. By using similarity-weighted voting, we can ensure that the nearest neighbors (which are most likely to have the same conditional probability) have the most significant effect on deciding $\mathbf{x}_j$'s label while those points with less similarity to $\mathbf{x}_j$ are not excluded entirely but have a weaker effect on labelling $\mathbf{x}_j$.

Algorithm 1 LatentMap: Transfer Learning between High Dimensional Overlapping Distributions

```
Input: $(X^\ell, Y^\ell), X^u$
Output: Labels of $X^u$
Build regression models $M = \{M_1, \ldots, M_{|F_+|}\}$ using $X^\ell$.
Use $M$ to predict missing values in $X^u$ and obtain $\hat{X}^u$
Construct $A \in \mathbb{R}^{t \times d}$ by taking $X^\ell$ as $A$'s first $\ell$ rows and $\hat{X}^u$ as next $u$ rows.
Apply SVD on $A$, $A = U\Sigma V^{\mathrm{T}}$.
Present the out-domain data in latent space, $V_{k-1} = A^{\mathrm{T}} U_{k-1} \Sigma_{k-1}^{-1}$
for Each in-domain instance $v_i, i = \ell+1, \ldots, \ell+u$ in latent space do
    Calculate the similarity between $v_i$ and all the out-domain instance $v_j, j = 1, \ldots, \ell$
    Retrieve top $p$ out-domain instances based on the calculated similarity.
    Label $\mathbf{x}_i$ by voting using the retrieved instances' labels.
end for
```

3. FORMAL ANALYSIS

In this section, we provide the theoretical basis for LatentMap. In transfer learning, the distributions of the two domains are different. For convenience, let 0 and 1 be the subscripts denoting out-domain and in-domain, respectively. We assume data from two domains are generated according to two unknown distributions $p_0(\mathbf{x}, y) = p_0(\mathbf{x})p_0(y|\mathbf{x})$ and $p_1(\mathbf{x}, y) = p_1(\mathbf{x})p_1(y|\mathbf{x})$, where $p_i(\mathbf{x}), i = 0, 1$ are the marginals, $p_i(y|\mathbf{x}), i = 0, 1$ are the conditionals, and $p_0(\mathbf{x}) \neq p_1(\mathbf{x})$,$p_0(y|\mathbf{x}) \neq p_1(y|\mathbf{x})$. It is difficult for learning algorithms to learn effectively since they usually assume $p_0(\mathbf{x}, y) = p_1(\mathbf{x}, y)$. The pivot is to find ways to mitigate the problem arising from this difference.

The outline of the analysis is as follows. First, we show that missing value prediction and participation of the in-domain data in SVD computation allow us to establish a bound for $|p_0(\mathbf{x}) - \hat{p}_1(\mathbf{x})|$ in latent space, where $\hat{p}_1$ represents the induced in-domain marginal in latent space. Next, under the clustering assumption [1] and by the bounded difference between the two marginals, we can further assume that the difference between the two conditionals is also bounded or at least similar across the two domains. Thus we can show that SVD not only brings two marginal distributions closer but also helps us discover clustering structures, where out-domain instances have their conditionals similar to a given in-domain instance. Note that we measure marginal distribution discrepency using Euclidean distance, and the Parzen windows method justifies this:

$$p_n(\mathbf{x}) = \frac{1}{\alpha_n} \sum_{i=1}^{n} \exp^{-\frac{(\mathbf{x}-\mathbf{x}_i)^2}{2\sigma^2}} \tag{8}$$

where $p_n(\mathbf{x})$ is the estimated density at $\mathbf{x}$, given instances $\mathbf{x}_i, i = 1, \cdots, n$. Thus $p_n(\mathbf{x})$ is the sum of n Gaussians centered at $\mathbf{x}_i$. $p_n(\mathbf{x})$ is a function of the distance between $\mathbf{x}$ and the instances $\mathbf{x}_i$. In the sequel, we shall consider the marginal distribution $p(\mathbf{x})$ as the empirical estimated marginal distribution $p_n(\mathbf{x})$ given by Equation (8).

3.1 Generalization Error with different Training and Test Distributions

As stated before, the challenge of transfer learning is that the joint distributions $p(\mathbf{x}, y)$ of the two domains are different. LatentMap aims at bridging this difference by bringing $\hat{p}_1(\mathbf{x})$ closer to $p_0(\mathbf{x})$. This strategy is justified by a theorem given in [16] that states that when the joint probabilities $p(\mathbf{x}, y)$ of the training and test data are different, the generalization error can be bounded asymptotically and the bound has two terms: one bounds the out-domain generalization error and the other bounds the difference between the two distributions. Assume that from the training data X^ℓ, Y^ℓ we can obtain a Bayesian predictive model

$$p(y|\mathbf{x}, X^\ell, Y^\ell) = \int p(y|\mathbf{x}, \omega) p(\omega|X^\ell, Y^\ell) d\omega \tag{9}$$

where ω is the parameter of the model. For $i = 0, 1$, let

$$G^i(\ell) = E^i_{\mathbf{x},y} E^0_{X^\ell, Y^\ell} [\log \frac{p_i(y|\mathbf{x})}{p(y|\mathbf{x}, X^\ell, Y^\ell)}] \tag{10}$$

Note that the expectation $E^i_{\mathbf{x},y}$ is over $\mathbf{x}$ and y with distribution $p_i(y|\mathbf{x})p_i(\mathbf{x})$. Thus $G^0(\ell)$ and $G^1(\ell)$ correspond to the generalization error with and without domain distributional difference, respectively. To give the generalization bound, two assumptions are made in [16]: (**A**) $G^i(\ell)$ has an asymptotic expansion and $G^i(\ell) \rightarrow B_i$ as $\ell \rightarrow \infty$, where B_i is a constant. (**B**) The largest difference between the training and test distributions is finite, i.e.

$$\mathcal{M}_0 = \max_{\mathbf{x}, y \sim p_0(y|\mathbf{x}) p_0(\mathbf{x})} [\frac{p_1(y|\mathbf{x}) p_1(\mathbf{x})}{p_0(y|\mathbf{x}) p_0(\mathbf{x})}] < \infty \tag{11}$$

Theorem 3.1 *Under the assumptions (*$\mathbf{A}$*) and (*$\mathbf{B}$*), the generalization error* $G^1(\ell)$ *asymptotically has an upper bound,*

$$G^1(\ell) \leq \mathcal{M}_0 G^0(\ell) + D_1 + D_2, \tag{12}$$

where

$$D_1 \quad = \quad \int p_1(y|\mathbf{x}) p_1(\mathbf{x}) \log \frac{p_1(y|\mathbf{x})}{p_0(y|\mathbf{x})} d\mathbf{x} dy$$

$D_2 = 0$ *if* $p_1(y|\mathbf{x}) = p_0(y|\mathbf{x})$ *and* 1 *otherwise.*

The detail of the proof can be found in [16]. $G^0(\ell)$ represents the out-domain generalization bound. D_2 is 1 since $p_0(y|\mathbf{x}) \neq p_1(y|\mathbf{x})$. To minimize the asymptotically generalization error upper bound in transfer learning, we want to minimize the rest two terms: D_1 and $\mathcal{M}_0$ relying on how close the two domain distributions are. First, D_1 is KL-divergence between the two conditionals $p_0(y|\mathbf{x})$ and $p_1(y|\mathbf{x})$. Second, $\max_{\mathbf{x}, y \sim p_0(y|\mathbf{x}) p_0(\mathbf{x})} p_1(\mathbf{x})/p_0(\mathbf{x})$ will be minimized when the difference between the induced in-domain marginal $\hat{p}_1(\mathbf{x})$ and $p_0(\mathbf{x})$ (fixed) can be minimized.

Empirically, given a finite number of in-domain points $\{\mathbf{x}_{\ell+1}, \dots, \mathbf{x}_{\ell+u}\}$ generated according to $p_1(\mathbf{x}, y)$, if the two conditional distributions are similar, then for all $i = \ell + 1, \dots, \ell+u$, $p(y|\mathbf{x}_i)$ will deviate from $p(y|\mathbf{x}_i)$ only by a small amount. Thus the estimated D_1 will be small. In addition, given out-domain points $X^\ell = \{\mathbf{x}_1, \dots, \mathbf{x}_\ell\}$, if in-domain points X^u are close to X^ℓ, according to the Parzen windows method, the estimated marginals $p_1(\mathbf{x}_i)$ and $p_0(\mathbf{x}_i)$ at $\mathbf{x}_i, i = 1, \dots, \ell$ as a function of the distance between $\mathbf{x}_i$ and points in X^ℓ and X^u, respectively, will also be close. We conclude that if the difference between the two distributions can be minimized, then the upper bound of generalization error can be minimized.

3.2 Missing Value Prediction via Regression

In this section, we analyze the effect of missing value prediction via regression. Given out-domain samples $\mathbf{x}_i \in X^\ell$ and an in-domain instance $\mathbf{x}_j \in X^u$, $\mathbf{x}_j$'s values along features F_+ are missing (treated as zeros), while $\mathbf{x}_i$ have values along features $F_+ \cup F_c$. Assume that $\hat{\mathbf{x}}_j$ is the same as $\mathbf{x}_j$ except having its values along F_+ predicted via regression models trained using X^ℓ. $\mathbf{x}$ can be projected onto F_+ and F_c, respectively and be written as $\mathbf{x} = ((\mathbf{x}^a)^t, (\mathbf{x}^b)^t)^t$, where $\mathbf{x}^a$ is the projection of $\mathbf{x}$ on F_+ and $\mathbf{x}^b$ is the projection on F_c. The squared distance between $\mathbf{x}$ and $\mathbf{x}_j$ is given by

$$D(\mathbf{x}, \mathbf{x}_j) = \|\mathbf{x} - \mathbf{x}_j\|^2 = \|\mathbf{x}^a - \mathbf{x}_j^a\|^2 + \|\mathbf{x}^b - \mathbf{x}_j^b\|^2 \qquad (13)$$

We assume that the difference between the two marginals over F_c is bounded, i.e. $\exists \mathcal{M}_1 > 0$ such that $\forall \mathbf{x}_i \in X^\ell, \mathbf{x}_j \in X^u, \|\mathbf{x}_i^b - \mathbf{x}_j^b\|^2 < \mathcal{M}_1$. Thus the second term in Equation (13) is the same for both $D(\mathbf{x}_i, \mathbf{x}_j)$ and $D(\mathbf{x}_i, \hat{\mathbf{x}}_j)$. Consider the first term in $D(\mathbf{x}_i, \hat{\mathbf{x}}_j)$. When y_+^{lj} are predicted with SVRs, we have

$$\begin{aligned} \|y_+^{li} - y_+^{lj}\|^2 &= \| < \mathbf{w}, \mathbf{x}_i^b > - < \mathbf{w}, \mathbf{x}_j^b > \|^2 \\ &\le \|\mathbf{w}\|^2 \|\mathbf{x}_i^b - \mathbf{x}_j^b\|^2 < \|\mathbf{w}\|^2 \mathcal{M}_1 \end{aligned}$$

where $\|\mathbf{w}\|^2$ is minimized when training the SVR model. Since $\mathbf{x}_i^a = (y_+^{1i}, \cdots, y_+^{|F_+|i})^t$ and $\hat{\mathbf{x}}_j^a = (y_+^{1j}, \cdots, y_+^{|F_+|j})^t$, thus $\|\mathbf{x}_i^a - \hat{\mathbf{x}}_j^a\|^2$ is also bounded and minimized. That is, $\exists \delta > 0$ such that

$$\|\mathbf{x}_i^a - \hat{\mathbf{x}}_j^a\|^2 = \sum_{l=1}^{|F_+|} (y_+^{li} - y_+^{lj})^2 < \delta$$

and δ is minimized through the regression model. On the other hand,

$$\|\mathbf{x}_i^a - \mathbf{x}_j^a\|^2 = \sum_{l=1}^{F_+} (y_+^{li} - 0)^2 = \|\mathbf{x}_i^a\|^2$$

which is fixed depending on $\mathbf{x}_i$. Compared with the minimized $D(\mathbf{x}_i, \hat{\mathbf{x}}_j)$, $D(\mathbf{x}_i, \mathbf{x}_j)$ is bounded but not minimized. Thus we conclude here that the marginal distribution is minimized via regression.

3.3 Bound for Distributional Difference

Projecting the data onto a lower dimensional space (latent space S) is necessary since the dimensionality of W is high. In this section, we first prove that the difference between $p_0(\mathbf{x})$ and $p_1(\mathbf{x})$ can be bounded in S, given it is bounded in W. Then by the clustering assumption, we bound the difference between $p_0(y|\mathbf{x})$ and $p_1(y|\mathbf{x})$ in S. Once these two distributional differences are bounded, then by applying Theorem 3.1 we claim that the generalization error in transfer learning is bounded through the proposed regression and SVD-based dimension reduction strategy.

Bound the marginal distributions in latent space In the previous section, we have discussed how the marginal distribution difference can be bounded in the space W. In this section, we further investigate properties of SVD and show how the difference between the two marginals can be bounded in latent space S, given this difference is bounded in W. In step 7 in Algorithm 1, the data matrix A in W is mapped to S by $V_{k-1} = A^{\mathrm{T}} U_{k-1} \Sigma_{k-1}^{-1}$ where $U_{k-1} = (\mathbf{u}_1, \dots, \mathbf{u}_{k-1})$. The mapping is $T = U_{k-1}\Sigma_{k-1}^{-1} : W \to S$, a point $\mathbf{x} \in W$ is mapped to $T\mathbf{x} \in S$ that is called $\mathbf{x}$'s image in S. S can be seen as a space spanned by basis $(\mathbf{u}_1/\sigma_1, \dots, \mathbf{u}_{k-1}/\sigma_{k-1})$. The following theorem concludes that we can further bound the marginal distributions of two domains by the mapping T.

Theorem 3.2 *Let $\mathbf{x}, \mathbf{x}'$ be two vectors in W and $T\mathbf{x}, T\mathbf{x}'$ be their images in the latent space S under the mapping $T = U_{k-1}\Sigma_{k-1}^{-1} : W \to S$. If $\|\mathbf{x} - \mathbf{x}'\|_2 < \delta, \delta > 0$, then $\|T\mathbf{x} - T\mathbf{x}'\|_2 < \delta\sqrt{\sum_{j=1}^{k-1} \frac{1}{\sigma_{k-1}^2}}$*

PROOF. First, each column of T, $T_j, j = 1, \dots, k-1$ has norm $\|T_j\|_2 = \sqrt{\sum_{i=1}^{t} |(T)_{ij}|^2} = \sqrt{\|u_j\|_2^2/\sigma_j^2} = 1/\sigma_j$. Second, the *Frobenius* norm the the linear transform T can be expressed as follow:

$$\begin{aligned} \|T\|_2^2 &= \|U_{k-1}\Sigma_{k-1}^{-1}\|_2^2 = \sum_{j=1}^{k-1}\sum_{i=1}^{t} |(T)_{ij}|^2 \\ &= \sum_{j=1}^{k-1}(\sum_{i=1}^{t} |(T)_{ij}|^2) = \sum_{j=1}^{k-1} \|T_i\|_2^2 = \sum_{j=1}^{k-1} \frac{1}{\sigma_i^2} \end{aligned}$$

Now we are ready to bound the distance of two images $T\mathbf{x}, T\mathbf{x}'$ in S.

$$\|T\mathbf{x} - T\mathbf{x}'\|_2^2 = \|T(\mathbf{x} - \mathbf{x}')\|_2^2 \le \|T\|_2^2 \|\mathbf{x} - \mathbf{x}'\|_2^2 < \delta^2 \sum_{j=1}^{k-1} \frac{1}{\sigma_j^2}$$

thus we have $\|T\mathbf{x} - T\mathbf{x}'\|_2 < \delta\sqrt{\sum_{j=1}^{k-1} \frac{1}{\sigma_j^2}}$ □

Basically, since we only choose the top $k - 1 (k <= 10)$ eigenvectors, we can use only those eigenvectors whose corresponding eigenvalues are larger than 1. Thus the distance of marginal distributions of two domains can be bounded in latent space.

Bound the conditional distributions in latent space In this section, we show that under the clustering assumption [12], the proposed retrieval strategy is optimal in making the conditional distributions $p_0(y|\mathbf{x})$ and $p_1(y|\mathbf{x})$ similar. Then in the next section, we derive the Bayes risk of this retrieval process. Following [12], the clustering assumption states that nearby points tend to have the same label. More precisely, let $\eta(\mathbf{x})$ be a regression function of y on $\mathbf{x}$, $\eta(\mathbf{x}) = p(y = 1|\mathbf{x})$, and $I(\cdot)$ be the indicator function. Then cluster assumption can be written as **(C)** Let $C_i, i = 1, \dots, k$ be clusters, then the function $\mathbf{x} \in X \to I(\eta(\mathbf{x}) \ge 1/2)$ takes a constant value on each of $C_i, i = 1, \ldots, k$. Alternatively, the above assumption is equivalent to $p(y = y'|\mathbf{x}, \mathbf{x}' \in C_i) \geq p(y \neq y'|\mathbf{x}, \mathbf{x}' \in C_i)$. So points in each cluster have the same $p(y|\mathbf{x})$. This is similar to the manifold assumption made in [1]. The assumption requires that $\eta(\mathbf{x})$ vary smoothly on the support of $p(\mathbf{x})$ which is a compact manifold, i.e. $\eta(\mathbf{x})$ should not vary significantly in a small enough area on the manifold. LatentMap follows these assumptions. In reality, however, the assumptions may not hold exactly, so we employ an instance retrieval strategy to approximate the cluster structures. For each in-domain instance $\mathbf{x}$, p nearest neighbors from the out-domain are retrieved. These neighbors are most likely to be in the same cluster as $\mathbf{x}$. Weighted voting is used to predict the label of $\mathbf{x}$, where a closer neighbor has more impact on deciding the label of $\mathbf{x}$. This is consistent with the manifold assumption that $\eta(\mathbf{x})$ of two points should be close when they are nearby. We conclude that the conditional distributions $p(y|\mathbf{x})$ of two domains are brought close within each cluster.

3.4 Upbound the Risk of Nearest Neighbor Classifier across Domains

In previous section, we show that the generalization error of transfer learning can be bounded. Since k-nn is employed as classifier in latent space S, we further analyze the Bayesian risk of k-nn in LatentMap. We conclude that the risk can be bounded and the upper bound can be minimized when two conditional distributions of the two domains are positive correlated. Assume that the marginal distributions $p_i(\mathbf{x}), i = 0, 1$ are continuous and are measurable with respect to a σ-finite measure ν. Next we show that for any in-domain instance $\mathbf{x}$, the nearest neighbor of $\mathbf{x}$ in the out-domain converges to $\mathbf{x}$ with probability one. We need some assumptions: (**D**) Both in-domain and out-domain data lie in the same space. This is true since both in-domain and out-domain data are in the latent space S. (**E**) Let $B_{\mathbf{x}}(r), r > 0$ be the ball $\{\hat{\mathbf{x}} \in X : d(\mathbf{x}, \hat{\mathbf{x}}) \leq r\}$ centered at $\mathbf{x}$ with radius r. $B_{\mathbf{x}}(r)$ has non-negative probability measure, $\forall r > 0$, with respect to the in-domain marginal probability.

Lemma 3.1 *Let in-domain instance* $\mathbf{x}$ *be drawn according to* $p_1(\mathbf{x})$ *and out-domain instances* $\mathbf{x}_1, \mathbf{x}_2, \ldots$ *be drawn according to* $p_0(\mathbf{x})$. *These instances are independent. Let* $\mathbf{x}'_\ell$ *be the nearest neighbor to* $\mathbf{x}$ *from the set* $\{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$. *Then* $\mathbf{x}'_\ell \to \mathbf{x}$ *with probability one.*

Proof. From the second assumption, for a fixed in-domain point $\mathbf{x} \in X$, for any $\delta > 0$, if the distance between $\mathbf{x}$ and the nearest neighbor $\mathbf{x}'_\ell$ from the out-domain samples $\{\mathbf{x}_1, \cdots, \mathbf{x}_\ell\}$ is larger than δ, then all the points in $\{\mathbf{x}_1, \cdots, \mathbf{x}_\ell\}$ are outside the sphere $B_{\mathbf{x}}(\delta)$, i.e.,

$$p_1\{d(\mathbf{x}, \mathbf{x}'_\ell) \geq \delta\} = (1 - p_1(B_{\mathbf{x}}(\delta)))^\ell \to 0$$

Consider a series of point sets: $P_\ell = \{\mathbf{x}_1, \cdots, \mathbf{x}_\ell\}$, with increasing ℓ, $d(\mathbf{x}, \mathbf{x}'_\ell)$ is monotonically decreasing. So the nearest neighbor of $\mathbf{x}$ converges to $\mathbf{x}$ with probability one. □

To find out the bound, we investigate the Bayes decision risk in the transfer learning setting. We define the loss function as 0-1 loss: $L(i, j) = 1$, if $i = j$ and 0 otherwise. Under such settings, the Bayes decision rule is

$$\begin{aligned} r^* &= \min_j \sum_{i=0}^{1} p(y|\mathbf{x})L(i, j) = \min\{p(y = 0|\mathbf{x}), p(y = 1|\mathbf{x})\} \\ &= \min\{p(y = 0|\mathbf{x}), 1 - p(y = 0|\mathbf{x})\} \end{aligned}$$

The Bayes decision rule minimizes the Bayes risk R^*, defined as

$$R^* = E[r^*] = \int r^* f(\mathbf{x}) dx$$

where $f(\mathbf{x}) = \sum_{i=0}^{1} p(y = i)p(\mathbf{x}|y = i)$. We have the following theorem that is the counterpart of the analysis of k-nn in [3] in the transfer learning setting.

Theorem 3.3 *Let* $p(\cdot|y = i), i = 0, 1$ *be such that with probability one,* $\mathbf{x}$ *is either 1) a continuity point of* $p(\cdot|y = i)$, *or 2) a point of non-zero probability measure. Then the risk* R *(probability of error) is bounded as*

$$\max\{R_1^*, R_0^*\} \leq R \leq R_1^* + R_0^* - 2R_1^*R_0^* - 2\mathbf{cov}(r_1^*, r_0^*).$$

where $R_i^*, i = 0, 1$ *is the out-domain and in-domain Bayes risk, respectively.*

Proof. For a fixed in-domain instance $(\mathbf{x}, y)$, let $(\mathbf{x}'_\ell, y'_\ell)$ be the nearest neighbor of $\mathbf{x}$ in the out-domain, where y and y'_ℓ are the labels of $\mathbf{x}$ and $\mathbf{x}'_\ell$, respectively. y and y'_ℓ are independent. Then the risk of misclassifying $\mathbf{x}$ is given by

$$\begin{aligned} r(\mathbf{x}, \mathbf{x}'_\ell) &= E[L(y, y'_\ell)|\mathbf{x}, \mathbf{x}'_\ell] = p(y \neq y'_\ell|\mathbf{x}, \mathbf{x}'_\ell) \\ &= p_1(y = 0|\mathbf{x})p_0(y'_\ell = 1|\mathbf{x}'_\ell) \\ &\quad + p_1(y = 1|\mathbf{x})p_0(y'_\ell = 0|\mathbf{x}'_\ell) \end{aligned}$$

Similar to [3], here we wish to show that $r(x, x'_\ell)$ converges to the random variable $r_1^* + r_0^* - 2r_1^*r_0^*$ with probabilities 1.

By Lemma 3.1 and the continuity of $p(\cdot|y = i)$, with probability 1, $p(\cdot|\mathbf{x}'_\ell) \to p(\cdot|\mathbf{x})$ for both in-domain and out-domain posterior probabilities. Thus

$$\begin{aligned} r(\mathbf{x}, \mathbf{x}'_\ell) &\to r^*(\mathbf{x}) \\ &= p_1(y = 0|\mathbf{x})p_0(y = 1|\mathbf{x}) \\ &\quad + p_1(y = 1|\mathbf{x})p_0(y = 0|\mathbf{x}) \\ &= p_1(y = 0|\mathbf{x})(1 - p_0(y = 0|\mathbf{x})) \\ &\quad + (1 - p_1(y = 0|\mathbf{x}))p_0(y = 0|\mathbf{x}) \end{aligned}$$

Since $r_1^* = \min\{p_1(y = 0|\mathbf{x}), 1 - p_1(y = 0|\mathbf{x})\}$ and $r_0^* = \min\{p_0(y = 0|\mathbf{x}), 1 - p_0(y = 0|\mathbf{x})\}$, $r^*(\mathbf{x})$ can be expressed as $r(\mathbf{x}) = r_1^*(1 - r_0^*) + (1 - r_1^*)r_0^*$. Overall risk $R = \lim_{\ell \to \infty} E[r(\mathbf{x}, \mathbf{x}'_\ell)]$. Because $r(\mathbf{x}, \mathbf{x}'_\ell)$ is bounded below 1, applying the dominated convergence theorem,

$$\begin{aligned} R &= E[\lim_{\ell \to \infty} r(\mathbf{x}, \mathbf{x}'_\ell)] = E[r(\mathbf{x})] && (14) \\ &= E[r_1^*(\mathbf{x})] + E[r_0^*(\mathbf{x})] - 2E[r_1^*(\mathbf{x})r_0^*(\mathbf{x})] && (15) \\ &= R_1^* + R_0^* - 2E[r_1^*(\mathbf{x})r_0^*(\mathbf{x})] && (16) \\ &= R_1^* + R_0^* - 2R_1^*R_0^* - 2\mathbf{cov}(r_1^*, r_0^*) && (17) \end{aligned}$$

where R_i^* is the Bayes risk that is the expectation of r_i^*. Rewriting Equation (14), we have

$$\begin{aligned} R &= E[r_1^*(\mathbf{x}) + r_0^*(\mathbf{x}) - 2r_1^*(\mathbf{x})r_0^*(\mathbf{x})] \\ &= R_1^* + E[r_0^*(\mathbf{x})(1 - 2r_1^*(\mathbf{x}))] \geq R_1^* \end{aligned}$$

similarly, we have $R \geq R_0^*$ thus we have

$$\max\{R_1^*, R_0^*\} \leq R \leq R_1^* + R_0^* - 2R_1^*R_0^* - 2\mathbf{cov}(r_1^*, r_0^*).$$

□

Since $r^* = \min\{p(y_1|\mathbf{x}), 1 - p(y_1|\mathbf{x})\}$, r_1^* and r_0^* can be positive correlated, giving a positive $\mathbf{cov}(r_{1)}^*, r_0^*)$.

Table 2: Data Summary

Data Sets	Instances		Features		$\lvert F_+\rvert/\lvert F_c\rvert$
	ℓ	u	$\lvert F_+\rvert$	$\lvert F_c\rvert$	
Re vs Si	2020	2008	1081	4172	0.2591
Au vs Av	2005	1980	810	4165	0.1945
C vs R	2431	1951	791	4345	0.1820
C vs S	2007	2373	682	5072	0.1345
C vs T	2218	1837	1007	5017	0.2007
R vs S	1963	1992	1017	4956	0.2052
R vs T	1885	1761	615	4677	0.1315
S vs T	1663	1939	490	5104	0.0960
O vs Pe	1239	1210	230	4091	0.0562
O vs Pl	1016	1046	178	3892	0.0457
Pe vs Pl	1079	1080	233	3834	0.0608

Remark *(1) If the two conditional distributions $p_0(y|\mathbf{x})$ and $p_1(y|\mathbf{x})$ are identical, the lower and upper bounds are the same as k-nn's (see [3]). Note that the upper bound can not be better than k-nn's upper bound. (2) In transfer learning, $p_0(y|\mathbf{x}) \neq p_1(y|\mathbf{x})$, the lower bound is the larger one of $\{R_0^*, R_1^*\}$, which indicates k-nn can not perform better when training and test data are from different domains than from a single domain. The upper bound says that if $p_0(y|\mathbf{x})$ and $p_1(y|\mathbf{x})$ are positively correlated, then the upper bound will be lower. However, when $p_0(y|\mathbf{x})$ and $y_1(y|\mathbf{x})$ negatively correlates, i.e. two domains' concepts contradict, then the upper bound grows. This is consistent with our intuition: transfer learning will benefit from two domains' similarity.*

3.5 Scalability Issues

One of the crux in LatentMap is to compute the SVD of a large matrix A. We don't have compute the exact SVD which requires $O(m^2n + mn^2)$ computational complexity where m and n are number of rows and columns respectively. Iterative algorithms for computing the first $k-1$ eigenvectors (and the corresponding eigenvalues) exist such as Lanczos method. Recently, randomized SVD are proposed, such as the method in [6], it sample columns of a large matrix according to a suitable probability distribution then a smaller matrix is constructed on which SVD is applied. This method provides a good approximation of the exact SVD and its running time is $O(mn + n)$. Another issue is find the top p nearest neighbor in ℓ out-domain instances for u in-domain data in the k dimension space, the computational complexity is $O(k * |\ell| * |u|)$.

4. EXPERIMENT

To demonstrate the effectiveness of the proposed framework, we carried out experiments on several data sets frequently used in transfer learning. Results show that LatentMap can map the data to a significant low dimension space where the distributions of two domains are similar. Missing values are dealt with properly, which improve the performance when the missing values are relatively copious.

4.1 Data Sets and Experiment Setup

We conducted experiments on three text data sets, all of which have different in- and out-domain distributions. We used SRAA (Simulated Real Auto Aviation), 20 newsgroups and Reuters-21758 as three main document classification tasks in this experiment. The SRAA corpus contains 73,218 UseNet articles from four discussion groups: simulated auto racing, simulated aviation, real autos, and real aviation. The 20 newsgroups corpus contains approximately 20,000 newsgroup documents, while Reuters-21758 contains 21758 Reuters news articles in 1987. Corpora are organized in a hierarchical manner. Our task is to classify documents into one of the top-level categories in the hierarchy. For example, in one of the 20 newsgroups task, we want to tell whether a document comes from category comp or rec. Since the distributions of in-domain and out-domain data are required to be different, we split documents from each top category into two sub-categories, one as the in-domain category and the other as the out-domain category. For example, the out-domain data consists of documents from comp.windows.x and rec.autos, while the in-domain data contains documents from comp.graphics and rec.motorcycles. All three main data sets are such organized that there are totally 11 transfer learning tasks in the experiment.

Raw text files are transformed into word vectors. All letters are turned into lower case and IDF-TF is used to produce term values. We discard terms whose occurrences are less than 2. Each word vector is normalized such that the length of the vector equals one. The Lovins stemming scheme is used to stem words appearing in the text. Simple tokening and stop word processing are used according to weka's default setting.

In cross-domain text classification problems, some features (terms) appear in both domains, while others appear only in the out-domain and missing in the in-domain, and vice verse. Table 2 shows the statistics of the features of all 11 tasks. As we can see, all the cross-domain tasks have missing values, i.e., F_+ is not empty. The last column in Table 2 shows the ratio $|F_+|/F_c$. In some tasks, this ratio is quite significant. For example, in the task of Real vs Simulated, $|F_+|$ is over one forth of $|F_c|$. In other tasks such as Orgs vs Places, the ratio is less than 1/20.

Baseline methods

We compare LatentMap with various learning algorithms, including naive Bayes (NB), Logistic regression (LR), decision trees (C4.5) and SVM. For the implementation of naive Bayes, Logistic regression and decision trees, we use the Weka package. SVMlight is used as SVM classifier. Procedural parameters are kept as default for all the classifiers. To predict missing values, we use SVR as our predictor and the implementation is provided by libSVM. The traditional learning algorithms assume that the training and test data are governed by an identical distribution $p(\mathbf{x}, y) = p(\mathbf{x})p(y|\mathbf{x})$. For this reason, we provide these learning methods with the original high-dimensional data. In particular, the union of F_+ and F_c are used as a whole and missing values are set to zero. We note that LatentMap has two key steps. First, the missing values in the in-domain data are predicted from the out-domain data. And as such, the induced in-domain marginal is closer to the out-domain marginal in input space W. Second, projecting the data onto a lower dimensional latent space S built using both the in- and out-domain data not only reveals cluster structures but also provides tight bounds for the two conditionals in the latent space. To see the effectiveness of these two steps, we include the following two methods in our comparison: 1) Running k-nn on data with regression but without dimension reduction (called k-nnReg, short for k-nn after Regression) 2) Mapping the data with missing values set to zero to latent space (called pLatentMap, short for partial LatentMap).

4.2 Performance Evaluation

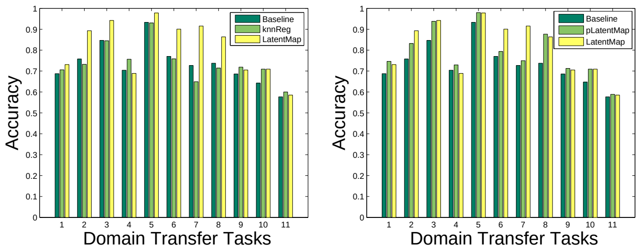
In this section, the experiment results of LatentMap against the baseline methods are provided. The results show clearly that LatentMap is able to bring two domains' joint distributions $p(\mathbf{x}, y)$ closer via regression and latent space projection, giving rise to a effective transfer learning framework.

Overall Performance Study
The results of LatentMap and other traditional methods on three data sets are summarized in Table 3 with the best results in bold font. It can be seen that in most of the tasks (10 out of 11) the performance is improved significantly. One exception is in task Comp vs Sci where the accuracy is slightly lower (within 3%) than the best of the baseline methods. Since all the baseline learning algorithms assume that the underlying distributions $p(\mathbf{x}, y)$ of training data and test data are identical, they perform poorly on most of the transfer tasks. For example, in the task Rec vs Talk from the 20 newsgroups data set, the lowest accuracy (around 60%) is achieved by Logistic regression and the decision tree, while the best learner (naive Bayes and SVMs) make correct predictions around 72%. In this situation, LatentMap outperforms the best baseline method with an improvement of near 20%. Over all, the smallest margin of improvement is around 2% on the task Orgs vs People.

Comparison of LatentMap and two simpler implementations is depicted in Figure 3. In Figure 3(a), we compare LatentMap and k-nnReg. It is clearly shown that by filling up the missing values, either LatentMap or k-nnReg outperforms the best of the baseline methods in 9 out of 11 tasks. Among these 9 tasks, LatentMap greatly outperforms k-nnReg in 6 tasks (task 2,3,5,6,7 and 8) and very close to k-nnReg in 2 tasks (task 10 and 11). This confirms that the second step that further discovers cluster structures in the latent space and moves the induced $p_1(y|\mathbf{x})$ closer to $p_0(y|\mathbf{x})$ can greatly improve the accuracy. Notice that while LatentMap has a lower accuracy than k-nnReg in tasks 10 and 11, it is still higher than that of the baseline methods.

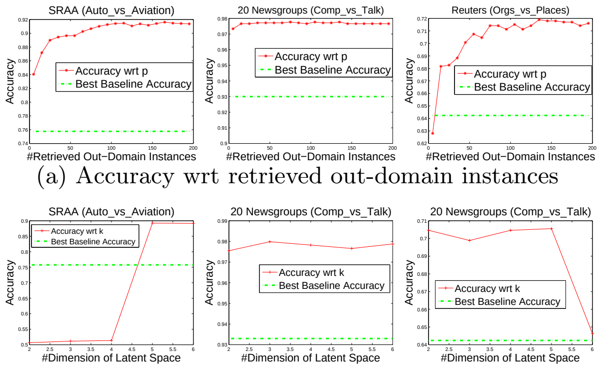
In Figure 3(b), we show the effect of multiple regression. By making the conditional distributions of two domains similar via latent space mapping, LatentMap and pLatentMap together outperform the baseline methods 9 out of 11 tasks (tasks 1,2,3,5,6,7,8,10 and 11). Furthermore, by predicting missing values through regression analysis, LatentMap outperforms pLatentMap in 5 out of these 9 tasks (task 2,3,6,7,11) with other three close to pLatentMap (tasks 8,9,10). Note that in Figure 3(b), on the last three tasks, LatentMap performs approximately the same as pLatentMap. By examining the last column of Table 2, we can see that LatentMap can achieve greater improvement on tasks where two domains overlap less or higher $|F_+|/|F_c|$ ratio (tasks Comp vs Talk and Rec vs Sci have the minimal feature set overlapping among six 20 Newsgroups tasks). Our results show that LatentMap can effectively lessen the discrepancy of two domains distributions. In particular, latent space mapping that discovers cluster structures can greatly improve the performance while regression analysis guarantees the performance when a large number of values is missing.

Parameter Sensitivity
There are two important parameters in the LatentMap algorithm: dimensionality of the latent space k and the number of documents in out-domain to retrieve for voting p. We choose one task from each of the three main tasks to examine the sensitivity issue. Since the distributions of in-domain and out-domain data are different, parameters chosen using cross-validation on the out-domain data will not work for the in-domain data.



(a) k-nnReg (b) pLatentMap

Figure 3: Effect of Two Key Steps



(a) Accuracy wrt retrieved out-domain instances

(b) Accuracy wrt dimensionality of latent space

Figure 4: Sensitivity Study

In this experiment, parameter p varies from 5 to 200 with increment 10 and k varies from 2 to 6 with increment 1. When p is changing, k is fixed at 5. The resulting accuracy curves are depicted in Figure 4(a). These accuracies are compared with that obtained from the traditional learning algorithm whose performance is the best. From the figure, it is obvious that the accuracy is improved as the number of retrieved out-domain instances increases, yet it remains stable after a certain threshold such as $p = 150$. This confirms the cluster assumption. That is, when more nearest neighbors are selected for voting, the effect of minor misclustered out-domain instances will be diminished or canceled out, leading to higher accuracy. The accuracy with respect to dimensionality of the latent space is higher than the best baseline classifier. Thus it is not critical which value k takes. When the underlying latent space's dimensions are changing, we always retrieve 50 instances from the out-domain, the resulting curves are shown in Figure 4(b)

5. RELATED WORK

One main challenge of transfer learning is how to resolve and, in the same time, take advantage of the difference between two domains. Some are based on instance weighting strategy ([2, 4, 7, 11]). For example, [4] adopts the boosting weight formula as the re-weighting scheme. Some other methods base on dimension reduction, which usually map

Table 3: Comparison of Performance

Methods	SRAA		20 News Groups						Reuters		
	Re vs Si	Au vs Av	C vs R	C vs S	C vs T	R vs S	R vs T	S vs T	O vs Pe	O vs Pl	Pe vs Pl
NB	0.6838	0.6889	0.8098	**0.7042**	0.89	0.7113	0.7189	0.704	0.6554	0.6424	0.5769
LR	0.6863	0.6768	0.8467	0.6195	0.933	0.7701	0.5928	0.6818	0.6471	0.6319	0.5046
C4.5	0.635	0.7576	0.6858	0.5908	0.7104	0.6391	0.5997	0.6266	0.5595	0.6195	0.5231
SVM	0.6877	0.7399	0.8401	0.6962	0.9107	0.7400	0.7269	0.7375	0.6860	0.6472	0.5250
LatentMap	**0.7311**	**0.8929**	**0.9421**	0.6890	**0.9777**	**0.9006**	**0.9154**	**0.8633**	**0.7050**	**0.7094**	**0.5852**

data to a new representation facilitating domain transfer ([9]). Recently, [8] proposes a locally weighted ensemble framework to combine multiple models for transfer learning by dynamically assigning weights of a model according to a model's predictive power on each test example. [10] proposes a information theory framework to address cross-language classification problem. [15] addressed the problem of cross-domain text classification using PLSA (Probabilistic Latent Semantic Analysis) to bridge domain transfer.

6. CONCLUSION

We address transfer learning challenges in text classification and other related problems, where the spaces of two domains are at most overlapping, the marginal and conditional distributions are different, and the dimensionality can be extremely high. We propose a framework (LatentMap) which draws joint distributions of two domains closer. The missing values are filled up to minimize the gap between marginal distributions, then the data is mapped to a latent space where both the marginal distribution and the relationship of two conditional distributions become easier to measure. Then, transferable sub-structures can be easily identified in the mapped low dimensional latent space. The dimensionality of the latent space is usually below 10, remarkably smaller as compared to the usual several thousands of the original space. Experiment over 11 text domain transfer tasks shows that LatentMap works as expected and achieves great improvement (up to around 20%) compared to traditional learning algorithms including SVM. Comparison with two simpler strategies (k-nnReg and pLatentMap) in the same transfer learning scenario shows that LatentMap can combine the advantages of both filling up missing value and latent space mapping. Parameters sensitivity analysis shows that LatentMap works well in very low dimensional spaces and is immune to the variation of the number of retrieved out-domain instances.

7. REFERENCES

[1] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434, 2006.

[2] Steffen Bickel, Michael Brückner, and Tobias Scheffer. Discriminative learning for differing training and test distributions. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 81–88, New York, NY, USA, 2007. ACM.

[3] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[4] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu. Boosting for transfer learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 193–200, New York, NY, USA, 2007. ACM.

[5] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 29, New York, NY, USA, 2004. ACM.

[6] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Mach. Learn.*, 56(1-3):9–33.

[7] Wei Fan and Ian Davidson. On sample selection bias and its efficient correction via model averaging and unlabeled examples. In *SDM*, 2007.

[8] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *Proceedings of the 2008 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[9] Xiao Ling, Wenyuan Dai, Gui-Rong Xue, Qiang Yang, and Yong Yu. Spectral domain-transfer learning. In *KDD '08: Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 488–496, New York, NY, USA, 2008. ACM.

[10] Xiao Ling, Gui-Rong Xue, Wenyuan Dai, Yun Jiang, Qiang Yang, and Yong Yu. Can chinese web pages be classified with english data source? In *WWW*, pages 969–978, 2008.

[11] Jiangtao Ren, Xiaoxiao Shi, Wei Fan, and Philip S. Yu. Type-independent correction of sample selection bias via structural discovery and re-balancing. In *SDM*, pages 565–576, 2008.

[12] Philippe Rigollet. Generalization error bounds in semi-supervised classification under the cluster assumption. *J. Mach. Learn. Res.*, 8:1369–1392, 2007.

[13] A. J. Smola and B. Schoelkopf. A tutorial on support vector regression, 1998.

[14] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[15] Gui-Rong Xue, Wenyuan Dai, Qiang Yang, and Yong Yu. Topic-bridged plsa for cross-domain text classification. pages 627–634. SIGIR, 2008.

[16] Keisuke Yamazaki, Motoaki Kawanabe, Sumio Watanabe, Masashi Sugiyama, and Klaus-Robert Müller. Asymptotic bayesian generalization error when training and test distributions are different. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 1079–1086, New York, NY, USA, 2007. ACM.