Source-Selection-Free Transfer Learning

Evan Wei Xiang[†], Sinno Jialin Pan[‡], Weike Pan[†], Jian Su[‡] and Qiang Yang[†]
[†]Department of Computer Science and Engineering,
Hong Kong University of Science and Technology, Hong Kong
[‡]Institute for Infocomm Research, 1 Fusionopolis Way, #21-01 Connexis, Singapore 138632
[†]{wxiang,weikep,qyang}@cse.ust.hk, [‡]{jspan,sujian}@i2r.a-star.edu.sg

Abstract

Transfer learning addresses the problems that labeled training data are insufficient to produce a high-performance model. Typically, given a target learning task, most transfer learning approaches require to select one or more auxiliary tasks as sources by the designers. However, how to select the right source data to enable effective knowledge transfer automatically is still an unsolved problem, which limits the applicability of transfer learning. In this paper, we take one step ahead and propose a novel transfer learning framework, known as source-selection-free transfer learning (SSFTL), to free users from the need to select source domains. Instead of asking the users for source and target data pairs, as traditional transfer learning does, SSFTL turns to some online information sources such as World Wide Web or the Wikipedia for help. The source data for transfer learning can be hidden somewhere within this large online information source, but the users do not know where they are. Based on the online information sources, we train a large number of classifiers. Then, given a target task, a bridge is built for labels of the potential source candidates and the target domain data in SSFTL via some large online social media with tag cloud as a label translator. An added advantage of SSFTL is that, unlike many previous transfer learning approaches, which are difficult to scale up to the Web scale, SSFTL is highly scalable and can offset much of the training work to offline stage. We demonstrate the effectiveness and efficiency of SSFTL through extensive experiments on several real-world datasets in text classification.

1 Introduction

Transfer learning aims to improve the learning performance in a target domain using knowledge extracted from related source domains or tasks. What distinguishes transfer learning from other traditional learning is that either the source and target domains, or the target and source tasks, or both, are different. Transfer learning is particularly useful when we only have limited labeled data in a target domain, which requires that we consult one or more auxiliary tasks or domains to gain insight on how to solve the target problem [Pan and Yang, 2010]. Many transfer learning approaches have been developed over the years. For example, [Raina *et al.*, 2006; Dai *et al.*, 2009] proposed to learn text classifiers by transferring knowledge from other text or image domains. [Pan *et al.*, 2010] and [Prettenhofer and Stein, 2011] proposed feature-based transfer learning methods for solving natural language processing tasks. [Ding *et al.*, 2011] proposed to adopt a boosting based approach to select the most discriminative feature for knowledge transfer in target domain.

In many typical transfer learning settings, a major assumption is that source data are provided by the problem designers. This places a big burden on the designer of the learning problem, since to improve the performance of learning, the "right" source data must be provided as well for effective transfer learning. However, it is very difficult to identify a proper set of source data. We often meet with the situation where we have a target task to solve, but we are at a loss at identifying from an extremely large number of choices of potential sources to use. For example, we may be given some text data to classify with limited labels, but we are only told to make use of the data on the World Wide Web! In such a situation, not only are we missing the source data, we also lack a scalable transfer learning method. This problem makes it difficult to benefit from many of the advantages of transfer learning.

In this paper, we propose a novel framework tap into the online information sources without asking the user for a specific source data set for a given target classification problem. For simplicity, we assume that the online information source and the target task share the same feature space, but the label spaces may be different or even disjoint. Our problem can be informally stated as follows. We are given a target text classification problem with categorical classes $\mathcal{Y}_T$. Besides the class label for the target classification problem, we optionally have a labeled training data that have a small number of class labels; however, we assume that the labels are not sufficient to build an effective classification model using traditional machine learning algorithms. We also have an entire information source $\mathcal{K}$ available online, such as the Wikipedia which is used in this paper. $\mathcal{K}$ consists of a collection of labeled text data, while the label space $\mathcal{Y}_S$ in $\mathcal{K}$ may be different from $\mathcal{Y}_T$ in the target data. To help bridge between $\mathcal{Y}_S$ and $\mathcal{Y}_T$, we also assume that we have a collection of social media that have been labeled via tags $\mathcal{Y}_{\text{tag}}$, such that $\mathcal{Y}_{\text{tag}}$

has overlap with $\mathcal{Y}_S$ and is a superset of $\mathcal{Y}_T$. In other words, the target labels are covered by the tags in $\mathcal{Y}_{\text{tag}}$, which in turn have overlaps of those labels in the online information source. Our goal is to build a "bridge" and select a subset of the $\mathcal{K}$ as the source data to transfer the knowledge for the target task.

To build an effective bridge between $\mathcal{Y}_S$ and $\mathcal{Y}_T$, we propose to embed all labels into a latent Euclidean space using a graph representation. As a result, the relationship between labels can be represented by the distance between the corresponding prototypes of the labels in the latent space. Furthermore, we show that predictions made by each source classifier can also be mapped into the latent space, which makes the knowledge transfer from source classifiers possible. Finally, we apply a regularization framework to learn an effective classifier for the target text classification task. In this manner, our transfer learning framework does not depend on the specification of a precise source data set by the problem designer, and for this reason we call it "source-selection-free transfer learning" (SSFTL for short).

There are several advantages associated with the SSFTL framework. First, since the online information source $\mathcal{K}$ is available ahead of the classifier training time for the target task. We can thus "compile" a large number of potential classifiers ahead of time for efficient classification, because they can be reused for different target tasks. Second, because we use a graph Laplacian to represent the label-translation process, the mapping between the target and online information source labels can be done very efficiently, resulting in a highly scalable architecture for transfer learning. Third, the class labels for the target learning task can vary from task to task, as long as they can be covered by the social media that serve as a bridge. This adds a lot to the flexibility of the learning process. Finally, our framework can be easily scaled up when the size of the online information source increases.

2 Source-Selection-Free Transfer Learning

2.1 Problem Definition

In the target domain we have a few labeled data $\mathcal{D}_T^\ell = \{(\boldsymbol{x}_i^\ell, y_i)\}_{i=1}^\ell$ and plenty of unlabeled data $\mathcal{D}_T^u = \{\boldsymbol{x}_i^u\}_{i=\ell+1}^n$, with label space $\mathcal{Y}_T$. In the auxiliary domain, we have k pre-trained classifiers $\{f^{S_i}\}$'s with label space $\mathcal{Y}_S = \bigcup_{i=1}^k \mathcal{Y}_{S_i}$. Both auxiliary and target domains have the same feature space, e.g. the same *bag of words* representations[1]. Assume that we have some social media e.g. Delicious, with tags $\mathcal{Y}_{\text{tag}}$ covering all labels in the target domain, $\mathcal{Y}_T \subset \mathcal{Y}_{\text{tag}}$. For simplicity in description, we also assume $\mathcal{Y}_S \subset \mathcal{Y}_{\text{tag}}$. Our goal is to learn a classifier in the target domain, by leveraging knowledge from the auxiliary classifiers and the social media data.

There are two challenges we need to address for the proposed problem, (1) since the label spaces of the auxiliary and target tasks may be different, a crucial research issue is how to build a bridge between these tasks via exploring relationships between the auxiliary and target labels, and (2) another challenge is how to make use of the pre-trained source classifiers to train a target classifier with the learned relationship.

[1]SSFTL can be extended for mismatched feature spaces by adopting various techniques introduced in [Pan and Yang, 2010]. Since this work mainly focuses on the problem of automatic source selection, we leave such extension in our future work.

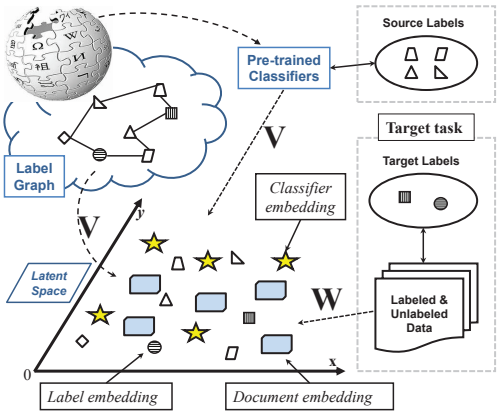

Figure 1: Source-Selection-Free Transfer Learning.

We address the above two challenges separately in our two-step transfer learning framework as shown in Figure 1. In the first step, we construct a label graph by utilizing the social tagging data, and then learn a latent low-dimensional space of the labels via graph embedding. In the second step, we propose a principled regularization framework to transfer the knowledge encoded in the source classifiers.

Notations Denote the union label set of the auxiliary and target labels $\mathcal{Y} = \mathcal{Y}_S \cup \mathcal{Y}_T$, where $|\mathcal{Y}| = q$, and $\phi(y) = [0, \ldots, 1, \ldots, 0]' \in \mathbb{R}^{q\times 1}$, where $y \in \mathcal{Y}$, a vector of dimensionality q with all zeros and a single 1 for the index of y in the label set $\mathcal{Y}$. We further define a vector representation of $f^{S_i}(x_j)$, $\boldsymbol{f}_j^{S_i} = [0, \ldots, p(y|x_j), \ldots, 0]' \in \mathbb{R}^{q\times 1}$, which is a vector of dimensionality q with all zeros and values $\{p(y|x_j)\}$'s at the indices of $\forall y \in \mathcal{Y}_{S_i}$, where $p(y|x_j)$ is the predicted conditional probability inferred from f^{S_i}. Note, for a given f^{S_i}, $\sum_{y\in\mathcal{Y}_{S_i}} p(y|x_j) = 1$, therefore $|\boldsymbol{f}_j^{S_i}| = 1$.

2.2 The SSFTL Algorithm

Label Graph Embedding

In this section, we show that the structure of the social tagging data can be exploited to extract the relationship between the target and auxiliary labels. Since the label names are usually short and sparse, it is very hard for us to identify their correspondence based on some similarity measure using their word feature space alone. In order to uncover the intrinsic relationships between the target and source labels, we turn to some social media such as Delicious, which can help to bridge different label sets together. Delicious can be viewed as a *Tag Cloud*, where different users may use different tags to label one Web page. Each tag can be treated as a label, and the tag co-occurrence relationship carries rich label correspondence information. In order to exploit the underlying structure of the graph in the social media data, we apply the graph spectral techniques [Chung, 1997] on the graph to map each node in the graph to a low-dimensional latent space. In this way, each label will have a coordinate on this latent space, and we call it the prototype of this class. Because the dimension of such latent space can be much lower than the original word feature space, the mismatch problem caused by the label sparseness can be alleviated. Then the relationships between labels, e.g., similar or dissimilar, can be represented by the distance between their corresponding prototypes in the latent space, e.g.,

close to or far away from each other.

Recall that $\mathcal{Y}$ is a union set of all source and target labels. For each label $y \in \mathcal{Y}$, we can find its corresponding category in the social media data $\mathcal{K}$. We can further extract a subgraph $\mathcal{G}$ that contains all target labels $\mathcal{Y}_T$ and the auxiliary labels $\mathcal{Y}_S$ from $\mathcal{K}$. For each label y, we aim to recover its low-dimensional representation $\boldsymbol{v}_y \in \mathbb{R}^{m\times 1}$. In this paper, we propose to use Laplacian Eigenmap [Belkin and Niyogi, 2003] to recover the latent matrix $\mathbf{V} = [\boldsymbol{v}_1, \dots \boldsymbol{v}_q]' \in \mathbb{R}^{q\times m}$.

Given a label graph $\mathcal{G}$ with its corresponding weight or neighborhood matrix $\mathbf{M} \in \mathbb{R}^{q\times q}$, Laplacian Eigenmap aims to learn $\mathbf{V}$ by solving the following optimization problem,

$$\min_{\mathbf{V}} \ \text{tr}(\mathbf{V}'\mathcal{L}\mathbf{V}) \ \text{ s.t. } \mathbf{V}'\mathbf{D}\mathbf{V} = \mathbf{I}_m, \tag{1}$$

where $\mathbf{D}$ is a diagonal matrix with $D_{ii} = \sum_j M_{ij}$, and $\mathcal{L} = \mathbf{D} - \mathbf{M}$ is the Laplacian matrix [Belkin and Niyogi, 2003]. $\mathbf{I}_m$ is an identity matrix of dimensionality m. Note that if the distance between two label prototypes across the auxiliary and target domains in the latent space is small, then it implies that these two labels are semantically similar. Thus, based on the distance between label prototypes in the latent space, we are able to transfer knowledge across domains.

Knowledge Transfer

Based on the discovery of the relationships between labels, we propose a principled regularization framework for source-selection-free transfer learning. In particular, with the matrix $\mathbf{V}$ estimated by using Laplacian graph embedding, for each label $y \in \mathcal{Y}$, we have its m-dimensional representation as $\boldsymbol{v}_y$. Therefore, for the target classification task which tries to learn a classifier $g : \boldsymbol{x} \to y$ with $y \in \mathcal{Y}_T$, we can transform it to a *regression* problem which aims at learning a regression model $g : \boldsymbol{x} \to \boldsymbol{v}_y$ with $\boldsymbol{v}_y = \mathbf{V}'\phi(y) \in \mathbb{R}^{m\times 1}$. In this paper, we assume g is a linear model which can be written as $g(\boldsymbol{x}) = \mathbf{W}'\boldsymbol{x}$ with $\mathbf{W} \in \mathbb{R}^{d\times m}$.

Recall that we are given a few target labeled data $\mathcal{D}_T^\ell$ and some target unlabeled data $\mathcal{D}_T^u$ in the target domain. In transfer learning, the labeled data are too few to learn a prediction model. We thus show how to use the unlabeled data in our framework for transfer learning.

We can make predictions on the unlabeled data $\mathcal{D}_T^u$ by using a combined mapping of all auxiliary classifiers as,

$$\mathbf{V}'\mathbf{F}_S^u = \mathbf{V}' \sum_{i=1}^{k} \varepsilon_i \mathbf{F}_{S_i}^u, \tag{2}$$

where $\mathbf{F}_{S_i}^u = [\boldsymbol{f}_{l+1}^{S_i}, \dots, \boldsymbol{f}_n^{S_i}] \in \mathbb{R}^{q\times(n-\ell)}$ is the predictions of auxiliary classifier $\boldsymbol{f}^{S_i}$ on $\mathcal{D}_T^u$, and $\{\varepsilon_i\}$'s are weights for the source classifiers $\{f^{S_i}\}$'s. We will introduce an effective approach to estimate ε_i at the end of this section.

The prototypes of the target labels may be enveloped and separated by those of the auxiliary labels in the latent space, which implies that the auxiliary classifiers may be helpful for the target regression problem. Therefore, the combined mapping of the auxiliary classifiers can be used to regularize the target classifier on the unlabeled target data as follows,

$$\Omega_{\mathbf{D}_T^u}(\mathbf{W}) = \frac{1}{n-\ell} \left\| \mathbf{W}'\mathbf{X}^u - \mathbf{V}'\mathbf{F}_S^u \right\|_F^2,$$

where $\mathbf{X}^u \in \mathbb{R}^{d\times(n-\ell)}$ is the unlabeled data matrix.

Finally, we obtain the following optimization problem,

$$\min_{\mathbf{W}} \Omega_{\mathcal{D}_T^\ell}(\mathbf{W}) + \lambda_1 ||\mathbf{W}||_F^2 + \lambda_2 \Omega_{\mathcal{D}_T^u}(\mathbf{W}), \tag{3}$$

where $\Omega_{\mathcal{D}_T^\ell} = \frac{1}{\ell} \left\| \mathbf{W}'\mathbf{X}^\ell - \mathbf{V}_T'\phi(\mathbf{Y}^\ell)) \right\|_F^2$ is the loss function on the labeled data matrix $\mathbf{X}^\ell \in \mathbb{R}^{d\times\ell}$ and $\phi(\mathbf{Y}^\ell) \in \mathbb{R}^{q\times\ell}$ is the corresponding label matrix. Such loss function can be replaced with some other large margin losses proposed in [Quadrianto *et al.*, 2010] or [Weinberger and Chapelle, 2009] which can penalize misclassification errors. The additional regularization term $||\mathbf{W}||_F^2$ is used to avoid overfitting. Note that the knowledge from the auxiliary classifiers is encoded in $\Omega_{\mathcal{D}_T^u}(\mathbf{W})$, and the relationships between the target and auxiliary labels are encoded in $\mathbf{V}$.

We show that the model parameter $\mathbf{W}$ in Eq.(3) can be solved analytically in the following proposition.

Proposition 1 *The optimization problem in Eq.(3) has an optimal solution in a closed form as*

$$\mathbf{W} = (\mathbf{A} + \lambda_1 \mathbf{I}_d)^{-1} \mathbf{X}\mathbf{F}'\mathbf{V}, \tag{4}$$

where $\mathbf{A} \succeq \mathbf{0}$ *is entirely independent of* $\mathbf{W}$, $\mathbf{X} = [\mathbf{X}^\ell \ \mathbf{X}^u] \in \mathbb{R}^{d\times n}$ *and* $\mathbf{F} = [\phi(\mathbf{Y}^\ell) \ \mathbf{F}_S^u] \in \mathbb{R}^{q\times n}$.

Due to space limit, we omit the proof of the proposition. If there is no labeled data available in the target task, i.e., $\ell = 0$, SSFTL reduces to an unsupervised learning model which only minimizes the second and the third terms in Eq.(3). It can be proved that in this case, the model parameter $\mathbf{W}$ still has a closed form solution, which is similar to Eq.(4).

So far, we have presented our SSFTL framework for transfer learning. We now introduce how to estimate the weights $\{\varepsilon_i\}$'s in Eq.(2). When there is no labeled data in the target domain, we can set $\varepsilon_i = 1/k$, which is a uniform weighting approach. However, if we have a few labeled data in the target domain, we can use the following simple yet effective approach to estimate $\{\varepsilon_i\}$'s. First, we can use each source classifier f^{S_i} to make predictions on the labeled target data $\mathcal{D}_T^\ell$ by using the following rule,

$$\arg\max_y P(y|\boldsymbol{x}_j^l) \propto -\left\| \mathbf{V}'\boldsymbol{f}^{S_i}(\boldsymbol{x}_j^l) - \mathbf{V}'\phi(y) \right\|_2^2.$$

We then calculate the classification accuracy h_i of each f^{S_i}, set $\varepsilon_i = h_i$ and normalize ε_i, s.t., $\sum_i \varepsilon_i = 1$. Such weights can help select the most useful knowledge to transfer.

Prediction

With the parameter matrix $\mathbf{W}$, we can make prediction on any incoming test data $\boldsymbol{x}$ using the following rule,

$$y^* = \arg\max_y P(y|\boldsymbol{x}) = \frac{e^{-||\mathbf{W}'\boldsymbol{x} - \boldsymbol{v}_y||_2^2}}{\sum_{y\in\mathcal{Y}_T} e^{-||\mathbf{W}'\boldsymbol{x} - \boldsymbol{v}_y||_2^2}}, \tag{5}$$

where the denominator is a normalization term, which ensures that $\forall y \in \mathcal{Y}_T, 0 \le P(y|\boldsymbol{x}) \le 1$ and $\sum_y P(y|\boldsymbol{x}) = 1$.

In practice, in learning the parameter matrix $\mathbf{W}$ in Eq.(4), we can apply linear conjugate gradient for estimating each column of $\mathbf{W}$ independently, without computing the inverse of a $d \times d$ matrix, which can be solved efficiently. Furthermore, the complexity of computing $\mathbf{F}_S^u$ is $O((n-l)dk)$. Since each source classifier is independent, we can further parallelize the computing process of $\mathbf{F}_S^u$, which can become much

more efficient. In making the prediction, the time complexity of SSFTL is $O(md|\mathcal{Y}_T|)$, which is independent of the number of auxiliary classifiers, and m is usually very small, e.g. $m = 100$ in our experiments, thus very efficient.

3 Experiments

3.1 Building Source Classifiers with Wikipedia

In this work, we incorporate Wikipedia as our online information source. Wikipedia is currently the largest knowledge repository on the Web. In our experiments, we downloaded the English Wikipedia mirror of August 10, 2009. In total, we got over 4 million pages, of which about 3 million pages are content articles, and around 0.5 million pages are categories organized in a directed graph structure. The entire corpus contains about 5.6 million words. We filtered out the words whose total term frequencies are lower than 100, and then obtained a relatively small dictionary with 200K words.

In order to train some source classifiers, we extracted a set of categories together with their content pages as labeled training data. Since a category in Wikipedia usually only contains around 50 content pages on average, we took the content pages belonging to its Tier-1 sub-categories as training instances as well. We further filtered out those categories containing fewer than 100 pages, or more than $5,000$ pages. Totally, we got $800,000$ pages with $10,000$ categories. We thus randomly sampled $50,000$ pairs of the categories to train binary classifiers. For each binary classifier, we used logistic regression. Finally, we got $50,000$ source binary classifiers. After obtaining the source classifiers, we no longer stored the training data (Wikipedia pages) for incoming learning tasks. Since there are up to 50,000 base classifiers, it would take about two days if we run the training process on a single server. Therefore, we distributed the training process to a cluster with 30 cores using MapReduce, and finished the training with two hours. These pre-trained source base classifiers are stored and reused for different incoming target tasks.

3.2 Building Label Graph with Delicious

As mentioned in the previous section, our proposed framework is based on a label graph over both the source and target label sets. In general, we do not know the label set of an incoming target task. If the pre-built label graph is too small to cover the target labels, we will have a low recall. One promising solution is to build a very large-scale label graph to reduce the probability that the incoming target label sets are not covered by the label graph. In this paper, we use a social tagging knowledge base, Delicious, to build the label graph. Delicious can be viewed as a *Tag Cloud*, where different users may use different tags to label one Web page. Each tag can be treated as a label, and the tag co-occurrence relationship carries rich label correspondence information. Thus, the learned latent vectors of the labels from the graph embedding have the property that categories which are semantically similar with each other will have similar vectors in the latent space, thus bridging the label space of two domains. We crawled 800-day historical tagging log from Delicious, ranging from January 1, 2005 to March 31, 2007. The data set contains about 50 million tagging logs of $200,000$ tags on 5 million Web pages, produced by 22 thousand users. We first aggregated the logs of individual users together, and then filtered out those low frequent tags and tagged pages. Finally, we obtained a bipartite graph consisting of $500,000$ page nodes, $50,000$ tag nodes, and 12 million links between the two types of nodes. Based on the bipartite graph, we constructed a tag neighborhood matrix $\mathbf{M}$ by setting the edge M_{ij} as the number of pages in which tag i and tag j co-occur. From Eq.(1), we can obtain the transformation matrix $\mathbf{V}$. In all experiments, the dimensionality of $\mathbf{V}$ is set to $m = 100$. The running time for building the neighborhood graph is less than one hour, and learning the Laplacian Eigenmap takes less than half an hour, due to the extremely sparsity of the matrix $\mathbf{M}$.

3.3 Target Tasks and Evaluation

For the target text classification tasks, we used the following real-world datasets. In order to ensure the source and target label sets are distinct, for each target dataset, we filtered the source classifiers whose corresponding label sets are overlapped with the target label set.

20NG: The 20 Newsgroups dataset[2] is a text collection of approximately $20,000$ newsgroup documents partitioned across 20 different newsgroups nearly evenly. We thus conducted 190 target binary text classification tasks, like *hockey vs. religion* and *medicine vs. motorcycles*.

Google: This dataset is about search snippets crawled from Google which consists of about $12,000$ labeled data under 8 categories. The detailed descriptions of the process can be found in [Phan *et al.*, 2008]. In total, we conducted 28 binary classification tasks.

AOL: This dataset is about queries from AOL provided by [Beitzel *et al.*, 2005] which contains $20,000$ labeled web queries under 17 categories[3]. Following the preprocess introduced in [Beitzel *et al.*, 2005], we enriched the queries with their top 50 search snippets. Finally, we got 136 binary classification tasks.

Reuters: AG corpus[4] is a collection of more than 1 million news articles which have been gathered from more than $2,000$ news sources by Antonio Gulli in more than 1 year. We selected the Reuters source which consists of 10,000 articles under 5 categories. We formed 10 binary classification tasks. Due to the limit of space, for each dataset, we only report an average result over all binary tasks in terms of classification accuracy. In all experiments, we set $\lambda_1 = 0.1$ in Eq.(3).

3.4 Experimental Results

To evaluate our proposed method for source-selection-free transfer learning, we first conduct some experiments to evaluate our proposed SSFTL method when a few labeled data in the target task are available. In this setting, we first compare our method with a Support Vector Machines (SVM) algorithm, which is applied on the target labeled data to train a binary classifier to make predictions on the target test data. The second baseline method is the Transductive Support Vector Machines (TSVM) algorithm, which is applied on the target labeled and unlabeled training data to learn a binary classifier to make predictions on the holdout target test data. In this experiment, for our method, we set $\lambda_2 = 0.01$ in Eq.(3), and the number of source classifiers to $5,000$. As shown in

[2] http://people.csail.mit.edu/jrennie/20Newsgroups/

[3] http://gregsadetsky.com/aol-data/

[4] http://www.di.unipi.it/∼gulli/AG_corpus_of_news_articles.html

Table 1: Comparison results under varying numbers of labeled data in the target task (accuracy in %).

Dataset	0		5			10			20			30		
	RG	SSFTL	SVM	TSVM	SSFTL	SVM	TSVM	SSFTL	SVM	TSVM	SSFTL	SVM	TSVM	SSFTL
20NG	50.0	**80.3**	69.8	75.7	**80.6**	72.5	81.0	**81.6**	79.1	83.7	**84.5**	83.7	84.9	**85.9**
Google	50.0	**72.5**	62.1	69.7	**73.4**	64.5	73.2	**75.7**	67.3	73.8	**80.3**	71.7	74.2	**81.1**
AOL	50.0	**71.0**	72.1	74.1	**74.3**	73.7	76.8	**77.7**	79.2	77.8	**80.7**	81.7	78.2	**82.5**
Reuters	50.0	**72.7**	69.7	63.3	**74.3**	75.9	63.7	**76.9**	79.5	66.7	**80.1**	81.8	69.8	**82.6**

Table 1, compared to SVM and TSVM, SSFTL can achieve much better classification accuracy on the target test data. An interesting result is that SSFTL can also achieve satisfiable classification performance without any labeled data, which is much higher than *Random Guessing* (RG).

Table 2: Comparison results on varying numbers of source classifiers (accuracy in %).

Dataset	Number of source classifiers for SSFTL						
	250	500	1K	2K	5K	10K	20K
20NG	76.3	78.2	80.3	82.5	84.5	85.1	**85.6**
Google	70.6	73.1	76.6	78.5	80.3	**80.4**	80.2
AOL	67.6	76.6	78.0	78.8	80.7	**81.2**	79.1
Reuters	72.2	74.0	76.7	78.0	**80.1**	79.6	78.1

In the second experiment, we aim to verify the impact of the number of source classifiers to the overall performance of SSFTL, where we set $\lambda_2 = 0.01$ and use 20 labeled target data. From Table 2, we can find that, when the number of source classifiers increases, the performance of SSFTL increases in company with the number. When it is equal to or larger than $5,000$, SSFTL can perform quite robustly and well. The reason may be that when the number of classifier increases, the number of source labels also increases. In this case, the prototypes of the target labels can be enveloped and separated by those of the source labels in the latent space with higher probability, which increases the chance for source-selection-free transfer learning possible.

Table 3: Comparison results on varying size of unlabeled data in the target task (accuracy in %).

Dataset	Unlabeled data involved in SSFTL				
	20%	40%	60%	80%	100%
20NG	80.5	80.9	81.8	84.0	**84.5**
Google	74.5	74.9	76.4	77.9	**80.3**
AOL	73.4	75.7	77.1	78.2	**80.7**
Reuters	75.5	77.7	77.8	78.7	**80.1**

In the third experiment, we further verify the performance of SSFTL when the proportion of unlabeled data involved in learning varies, as shown in Table 3. In this experiment, we use $5,000$ source classifiers, 20 labeled target data and set $\lambda_2 = 0.01$. The results suggest that the classification performance of SSFTL increases as the amount of unlabeled data grows. To be emphasized that unlabeled data are always cheap and easy to obtain. As a result, our proposed SSFTL can benefit from the semi-supervised setting.

Note that in the proposed SSFTL, besides λ_1 being fixed in all experiments, there exists another parameter λ_2 to be set. In the following experiment, we verify the impact of different values of λ_2 on the overall classification performance of SSFTL. The result is shown in Table 4. In this experiment, we use $5,000$ source classifiers and 20 labeled data. As can be seen, the proposed SSFTL performs best and is stable when λ_2 falls in the range $[0.001,\ 0.1]$. When $\lambda_2 = 0$, the semi-supervised SSFTL method is reduced to a supervised regularized least squares regression (RLSR) model, and when the value of λ_2 is large, e.g. $\lambda_2 = 100$, the result of SSFTL is similar to those of unsupervised SSFTL as shown in Table 1.

Table 4: Overall performance of SSFTL under varying values of λ_2 (accuracy in %).

Dataset	λ_2 of SSFTL						
	0	0.001	0.01	0.1	1	10	100
20NG	83.2	84.1	84.5	**85.3**	84.8	83.3	79.3
Google	76.6	79.1	**80.3**	78.7	78.2	77.4	74.3
AOL	78.3	79.5	**80.7**	79.1	78.8	76.3	73.4
Reuters	75.5	78.2	**80.1**	78.5	76.0	72.1	68.5

In the last experiment, we verify the effectiveness of our proposed weighted strategy of auxiliary source classifiers introduced at the end of Section 2. We compare the classification performance of SSFTL using the weighted strategy with that using the uniform weighting strategy. In this experiment, we set $\lambda_2 = 0.01$, use $5,000$ source classifiers and vary the number of labeled target data. As can be seen from Table 5, SSFTL using the weighted strategy can perform much better than that using the uniform weighting strategy. With this simple weighted strategy, we are able to "filter" unrelated source classifiers and identify useful ones for transfer.

Table 5: Analysis on weighted and uniform SSFTL under varying number of labeled data (accuracy in %).

Dataset	Uniform SSFTL				Weighted SSFTL			
	5	10	20	30	5	10	20	30
20NG	72.8	80.7	81.2	81.9	80.6	81.6	84.5	85.9
Google	64.1	67.0	70.8	77.2	73.4	75.7	80.3	81.1
AOL	69.8	71.7	72.1	74.8	74.3	77.7	80.7	82.5
Reuters	69.7	70.3	75.5	78.8	74.3	76.9	80.1	82.6

4 Related Works

Most previous works of transfer learning methods in text classification require the label spaces between the source and target tasks to be the same, and assume the difference between domains is only caused by the mismatch of data distributions [Pan and Yang, 2010]. In order to guarantee the scalability of large scale transfer learning, Duan *et al.*, proposed a domain adaptation machine (DAM) [Duan *et al.*, 2009] for transfer learning. Similar to SSFTL, in DAM, the knowledge carried by the source domains are encoded in the compact model parameters instead of the reuse of the raw data. Gao *et al.* [2008] proposed a locally weighted ensemble framework (LWE) to combine multiple models for transfer learning. However, either DAM or LWE needs to assume

the label spaces for the source and target tasks be the same, which cannot be applied to solve the label mismatch problem.

More recently, the label mismatch problem in instance-based transfer learning has attracted more and more attention. Some research works, such as risk-sensitive spectral partition (RSP) [Shi *et al.*, 2009], EigenTransfer [Dai *et al.*, 2009] and multi-task learning with mutual information (MTL-MI) [Quadrianto *et al.*, 2010], introduced some transfer learning methods for learning the label correspondence. However, their learning processes require maintaining all the training data from the auxiliary domain, which is ineffective for large scale setting if not impossible. As far as we know, SSFTL is the first work to address the above two challenges of heterogeneous label spaces and scalability due to large auxiliary domain, and we summarize them in Table 6.

There are also some recent works on label embedding [Weinberger and Chapelle, 2009; Bengio *et al.*, 2010] to discover a compressed space for large-scale multiple classes, such that a multi-class problem can be transformed to a regression problem. Our work is focused on exploring the relationships between the source and target labels to bridge two domains to enable knowledge transfer.

Table 6: Summary of some related transfer learning works.

Transfer learning methods	*Scalability*	*Diff. label*
RSP [Shi *et al.*, 2009]	$\times$	$\surd$
EigenTransfer [Dai *et al.*, 2009]	$\times$	$\surd$
MTL-MI [Quadrianto *et al.*, 2010]	$\times$	$\surd$
DAM [Duan *et al.*, 2009]	$\surd$	$\times$
LWE [Gao *et al.*, 2008]	$\surd$	$\times$
SSFTL	$\surd$	$\surd$

5 Conclusions and Future Work

In this paper, we proposed a novel transfer learning framework, known as source-selection-free transfer learning (SSFTL), to solve transfer learning problems when the potential auxiliary data is embedded in very large online information sources. In our SSFTL framework, the label sets across domains can be different. We compile the label sets into a graph Laplacian for automatic label bridging, such that model designers no longer need to select task-specific source-domain data. SSFTL is highly scalable because the processing of the online information source can be done offline and reused for different tasks. Extensive experiments have been conducted to verify that SSFTL is efficient and effective for transfer learning. In the future, we will extend SSFTL along the following directions: (1) extend SSFTL to achieve knowledge transfer on heterogeneous feature spaces; (2) generalize SSFTL to truly "source-free" via transferring knowledge with different forms from the World Wide Web.

6 Acknowledgements

Evan W. Xiang, Weike Pan and Qiang Yang thank the support of Hong Kong RGC/NSFC N_HKUST624/09 and Hong Kong RGC grant 621010.

References

[Beitzel *et al.*, 2005] Steven M. Beitzel, Eric C. Jensen, Ophir Frieder, David D. Lewis, Abdur Chowdhury, and Aleksander Kolcz. Improving automatic query classification via semi-supervised learning. In *ICDM*, 2005.

[Belkin and Niyogi, 2003] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[Bengio *et al.*, 2010] Samy Bengio, Jason Weston, and David Grangier. Label embedding trees for large multi-class tasks. In *NIPS 23*. 2010.

[Chung, 1997] Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

[Dai *et al.*, 2009] Wenyuan Dai, Ou Jin, Gui-Rong Xue, Qiang Yang, and Yong Yu. Eigentransfer: a unified framework for transfer learning. In *ICML*, 2009.

[Ding *et al.*, 2011] Wei Ding, Tomasz F. Stepinski, Yang Mu, Lourenco Bandeira, Ricardo Ricardo, Youxi Wu, Zhenyu Lu, Tianyu Cao, and Xindong Wu. Sub-kilometer crater discovery with boosting and transfer learning. *ACM TIST*, To appear 2011.

[Duan *et al.*, 2009] Lixin Duan, Ivor W. Tsang, Dong Xu, and Tat-Seng Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *ICML*, 2009.

[Gao *et al.*, 2008] Jing Gao, Wei Fan, Jing Jiang, and Jiawei Han. Knowledge transfer via multiple model local structure mapping. In *KDD*, 2008.

[Pan and Yang, 2010] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE TKDE*, 22(10):1345–1359, October 2010.

[Pan *et al.*, 2010] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feauture alignment. In *WWW*, 2010.

[Phan *et al.*, 2008] Xuan Hieu Phan, Minh Le Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW*, 2008.

[Prettenhofer and Stein, 2011] Peter Prettenhofer and Benno Stein. Cross-lingual adaptation using structural correspondence learning. *ACM TIST*, To appear 2011.

[Quadrianto *et al.*, 2010] Novi Quadrianto, Alex J. Smola, Tiberio S. Caetano, S.V.N. Vishwanathan, and James Petterson. Multitask learning without label correspondences. In *NIPS 23*, 2010.

[Raina *et al.*, 2006] Rajat Raina, Andrew Y. Ng, and Daphne Koller. Constructing informative priors using transfer learning. In *ICML*, 2006.

[Shi *et al.*, 2009] Xiaoxiao Shi, Wei Fan, Qiang Yang, and Jiangtao Ren. Relaxed transfer of different classes via spectral partition. In *ECML/PKDD*, 2009.

[Weinberger and Chapelle, 2009] Kilian Q Weinberger and Olivier Chapelle. Large margin taxonomy embedding for document categorization. In *NIPS 21*. 2009.