

Transfer Latent Semantic Learning: Microblog Mining with Less Supervision

¹Dan Zhang, ²Yan Liu, ³Richard D. Lawrence, ³Vijil Chenthamarakshan

¹Department of Computer Science, Purdue University, West Lafayette, IN 47907, US

²Department of Computer Science, University of Southern California, Los Angeles, CA 90089, US

³Machine Learning Group, IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, US

¹zhang168@cs.purdue.edu, ²yanliu.cs@usc.edu, ³{ricklawr, ecvijil}@us.ibm.com

Abstract

The increasing volume of information generated on microblogging sites such as Twitter raises several challenges to traditional text mining techniques. First, most texts from those sites are abbreviated due to the constraints of limited characters in one post; second, the input usually comes in streams of large-volumes. Therefore, it is of significant importance to develop effective and efficient representations of abbreviated texts for better filtering and mining. In this paper, we introduce a novel transfer learning approach, namely transfer latent semantic learning, that utilizes a large number of related tagged documents with rich information from other sources (source domain) to help build a robust latent semantic space for the abbreviated texts (target domain). This is achieved by simultaneously minimizing the document reconstruction error and the classification error of the labeled examples from the source domain by building a classifier with hinge loss in the latent semantic space. We demonstrate the effectiveness of our method by applying them to the task of classifying and tagging abbreviated texts. Experimental results on both synthetic datasets and real application datasets, including Reuters-21578 and Twitter data, suggest substantial improvements using our approach over existing ones.

Introduction

Micro-blogging sites such as Twitter allow users to send and receive short messages from other users. Twitter has become a novel real-time channel for people to share information on a broad range of subjects, such as personal updates, fast-breaking news, politics, entertainment, and just about anything else that people might discuss in everyday conversation. With at least 50 million messages (tweets) posted daily, there is an obvious need for effective ways to organize this information via relevance filtering, topic modeling, tracking trends (i.e., “hot” topics) and so on.

There are two major characteristics that distinguish microblogging data from traditional text corpus. First, each post or tweet is limited to 140 characters, and as a result abbreviated syntax is often introduced to accommodate the limit. Second, the input data typically arrives in high-volume streams. These characteristics raise great challenges to existing mining algorithms based on “bag-of-words” representation. It is extremely difficult to infer the context from short

texts only. For example, given tweets like “prize to BP for proving oil & water DO mix” and “Chemistry Prize awarded for disproving the old belief that oil and water don’t mix. BP is a recipient who couldn’t be with us tonight”, it is hard for machine learning algorithms to tag them as relevant to the BP oil-spill event. Since the texts come in streams and the topics might change quickly over the time, determining the contexts of words beforehand is impractical. Therefore, it is of great importance to develop effective and efficient representation of short texts for better filtering and mining.

In this paper, we propose transfer latent semantic learning (Transfer-LSI), which utilizes related information from more complete documents (such as wikipedia or socially tagged web pages) as supplementary information to infer a shared latent semantic space and hence bridge the gap between short texts and their implicit context. It is motivated by the observation that because humans read news, wiki, and blogs to accumulate the knowledge specific to a domain, they can better understand the short texts in microblogging. There are two major technical challenges in this approach: (1) what types of related sources are most appropriate for Transfer-LSI? (2) given the related information, how to build an effective latent space?

Transfer learning has been extensively studied in the past decade to leverage the data (either labeled or unlabeled) from one task (or domain) to help another (Pan and Yang 2009). In summary, there are two types of transfer learning problems, i.e., shared label space and shared feature space. For shared label space (Arnold, Nallapati, and Cohen 2008), the main objective is to transfer the label information between observations from different distributions (i.e., domain adaptation) or uncover the relations between multiple labels for better prediction (i.e., multi-task learning); for shared feature space, one of the most representative works is self-taught learning (Raina et al. 2007), which uses sparse coding (Lee et al. 2007) to construct higher-level features via abundant unlabeled data to help improve the performance of the classification task with a limited number of labeled examples. The problem addressed in this paper is closely related to self-taught learning, except that (i) the inputs from the target domain are short texts, i.e., extremely sparse feature vectors; (ii) there is no specific classification task at the end. The sparsity in features raises great challenges to unsupervised algorithms like self-taught learning because they

Latent Space	Example words
# 1	dir, flex, eclipse, adobe, google, swg, calendar, golf, flash, plugin
# 2	warranty, lenovo, look, foundation, flex, skin, wikicentre, label, emea, printable
# 3	interview, queue, wave, inquiry, twice, music, science, symphoni, citi, student

Table 1: Top three ranked latent space learned by self-taught learning. For each latent space, example words with the highest contributions are shown.

Latent Space	Example words
# 1	document, lotus, image, doc, mail, quickr, symphoni, oracle, sametime, domino
# 2	cookie, portlet, browser, alter, javascript, yahoo, opera, seller, page, workplace
# 3	blog, aim, aol, comment, feed, post, connect, rss, flickr, tag

Table 2: Top three ranked latent space learned by Transfer-LSI. The semantics for each latent space seems much more meaningful. For example, latent space # 1 is related to software, # 2 can be considered as web-related, while # 3 is more about RSS.

can only capture the dominant latent space and the results are directly determined by the unlabeled examples (i.e., the source domain). For example, Table 1 shows the latent semantic space learned by self-taught learning on a collection of webpages on “Lotus” software (detailed experiment settings can be found in Section 4). It is clear that these learned latent spaces can not be interpreted easily.

To achieve a more robust latent space, we propose using “labeled examples” (e.g. tagged documents) to guide the learning process. From a theoretical perspective, it has been shown that discriminative training of generative latent models, such as LDA or harmonium models, can achieve much robust latent spaces than pure generative models (Lacoste-Julien, Sha, and Jordan 2008). From a cognitive perspective, the concepts and most of their associated words would remain stable in different contexts or media. From a practical perspective, the number of tagged webpages has increased dramatically so we can access abundant tagged documents easily. Table 2 shows the latent semantic space learned from labeled examples by our algorithm. It can be observed that the resulting latent space is much more meaningful compared with self-taught learning.

Our proposed algorithm, namely transfer latent semantic learning, consists of two stages. First, in the source domain, we learn a latent semantic space from labeled examples by simultaneously achieving the best reconstruction of the source documents, and minimizing the error in predicting the known labels. Second, once the latent space is obtained, the examples in the target domain are mapped to this space, which can be used for future learning or mining tasks. Through this way, the knowledge in the source domain is transferred to the target domain.

Methodology

Problem Statement and Notations

Suppose we are given a dataset $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \in \mathcal{X}$, with extremely sparse features, as the target domain, as well as m examples $\{(\mathbf{z}_1, \mathbf{y}_1), (\mathbf{z}_2, \mathbf{y}_2), \dots, (\mathbf{z}_m, \mathbf{y}_m)\} \in \{\mathcal{X}, \mathcal{Y}\}$ in the source domain. \mathcal{X} is a d -dimensional space, and \mathcal{Y} denotes the label space. Here, without loss of generality, in this paper, we assume that different labels in the label space are mutually independent, and each \mathbf{y}_i can be considered as

a l -dimensional label vector, as in multi-label learning problems, with $\mathbf{y}_{ij} \in \{0, 1\}$ and l being the number of binary labels in the label space. The main purpose of Transfer-LSI is to infer a latent space that conveys some semantic meanings shared by both the source domain and the target domain so that the target domain examples can be reconstructed based on this latent semantic space.

Method

Transfer-LSI is a novel problem, and in this paper, we propose a general framework with two steps:

1. Latent Semantic Space Inference. The multi-labeled examples in the source domain are used to learn a higher level, more succinct representation of the inputs, i.e., a latent semantic space. For example, if both the source domain and target domain are text documents and each feature represents a specific word, the proposed method will learn a set of different word combinations that can comprise all of these documents in the source domain, and are consistent with their labels. In other words, it tries to discover the best higher level representation on the source domain that can perform both the classification and the feature reconstruction tasks well. The reason why the target domain examples are not used here is that there are too many missing features for the target domain examples, and therefore if we directly use them to get the latent semantic space, the result will be badly affected.
2. Target Domain Reconstruction. The proposed method represents the examples in the target domain in terms of the latent space obtained from the previous step. To handle the feature sparsity problem in the target domain, for each example, only the non-zero features are considered for reconstruction. In this step, the knowledge in the source domain is transferred to the target domain.

Latent Semantic Space Inference

The latent semantic space inference can be learned using a modified version of self taught learning, which was introduced in (Raina et al. 2007). More precisely, Transfer-LSI can be formulated as follows:

$$\begin{aligned}
\min_{\mathbf{W}, \mathbf{b}, \mathbf{A}, \Phi} \quad & C_1 \sum_{i=1}^m (\|\mathbf{z}_i - \Phi \mathbf{a}_i\|_2^2 + \beta \|\mathbf{a}_i\|_2^2) + \frac{1}{2} \sum_{j=1}^l \|\mathbf{w}_j\|^2 \\
& + C_2 \sum_{i=1}^m \sum_{j=1}^l \xi_{ij} \\
s.t. \quad & \forall i \in \{1, \dots, m\}, \forall j \in \{1, \dots, l\} \\
& \mathbf{y}_{ij} (\mathbf{w}_j^T \mathbf{a}_i + b_j) \geq 1 - \xi_{ij}, \quad \xi_{ij} \geq 0 \\
& \forall i \in \{1, \dots, s\}, \quad \|\Phi_i\|_2 \leq 1.
\end{aligned} \tag{1}$$

There are four sets of variables that need to be optimized in this formulation. $\Phi \in \mathbb{R}^{d \times s}$ is a s -dimensional hidden space underlying both the source and target domains. $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m]$ are the activation coefficients for $[\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m]$ in this hidden space. A set of multi-label classifiers $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_l]$ is trained based on \mathbf{A} , where $\mathbf{b} = [b_1, b_2, \dots, b_l]$ represents the corresponding biases.

The optimization problem contains two parts, the reconstruction part $C_1 \sum_{i=1}^m (\|\mathbf{z}_i - \Phi \mathbf{a}_i\|_2^2 + \beta \|\mathbf{a}_i\|_2^2)$ and the supervision part $\frac{1}{2} \sum_{j=1}^l \|\mathbf{w}_j\|^2 + C_2 \sum_{i=1}^m \sum_{j=1}^l \xi_{ij}$. C_1 , C_2 are trade-off parameters, while β is the regularization parameter for the reconstruction part. It is clear that this formulation tries to minimize the reconstruction error and the multi-label classification loss simultaneously. Although not jointly convex, this optimization problem is convex in \mathbf{W} , \mathbf{b} , Φ (while fixing \mathbf{A}), and convex in \mathbf{A} (while fixing \mathbf{W} , \mathbf{b} , Φ). Therefore, in this paper, we iteratively optimize this optimization problem by alternatively optimizing w.r.t. \mathbf{W} , \mathbf{b} , Φ and \mathbf{A} . In particular,

1. Fixing \mathbf{A} , $(\mathbf{W}, \mathbf{b}, \Phi)$ can be solved by the following convex optimization problem:

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{b}, \Phi} \quad & C_1 \sum_{i=1}^m \|\mathbf{z}_i - \Phi \mathbf{a}_i\|_2^2 + \frac{1}{2} \sum_{j=1}^l \|\mathbf{w}_j\|^2 + C_2 \sum_{i=1}^m \sum_{j=1}^l \xi_{ij} \\ \text{s.t.} \quad & \forall i \in \{1, \dots, m\}, \forall j \in \{1, \dots, l\} \\ & \mathbf{y}_{ij} (\mathbf{w}_j^T \mathbf{a}_i + b_j) \geq 1 - \xi_{ij}, \quad \xi_{ij} \geq 0 \\ & \forall i \in \{1, \dots, s\}, \quad \|\Phi_i\|_2 \leq 1. \end{aligned} \quad (2)$$

It is clear that the above optimization problem can be divided into two independent subproblems, i.e.,

$$\begin{aligned} \min_{\Phi} \quad & \sum_{i=1}^m \|\mathbf{z}_i - \Phi \mathbf{a}_i\|_2^2 \\ \text{s.t.} \quad & \forall i \in \{1, \dots, s\}, \quad \|\Phi_i\|_2 \leq 1, \end{aligned} \quad (3)$$

and for each \mathbf{w}_j , and b_j :

$$\begin{aligned} \min_{\mathbf{w}_j, \mathbf{b}} \quad & \frac{1}{2} \|\mathbf{w}_j\|^2 + C_2 \sum_{i=1}^m \xi_{ij} \\ \text{s.t.} \quad & \forall i \in \{1, \dots, m\}, \mathbf{y}_{ij} (\mathbf{w}_j^T \mathbf{a}_i + b_j) \geq 1 - \xi_{ij}, \quad \xi_{ij} \geq 0. \end{aligned} \quad (4)$$

Similar to (Lee et al. 2007), problem (3) can be solved efficiently by using the Lagrange dual. The Lagrangian dual for problem (3) can be formulated as follows:

$$\mathcal{L}(\Phi, \lambda) = \text{trace}((\mathbf{Z} - \Phi \mathbf{A})^T (\mathbf{Z} - \Phi \mathbf{A})) + \sum_{i=1}^s \lambda_i (\|\Phi_i\|_2^2 - 1), \quad (5)$$

where $\lambda_i \geq 0$ is the Lagrangian dual multiplier and $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m]$. The dual form of problem (3) is:

$$\begin{aligned} \mathcal{D}(\lambda) &= \min_{\Phi} \mathcal{L}(\Phi, \lambda) \\ &= \text{trace}(\mathbf{Z}^T \mathbf{Z} - \mathbf{Z} \mathbf{A}^T (\mathbf{A} \mathbf{A}^T + \Lambda)^{-1} (\mathbf{Z} \mathbf{A}^T)^T - \Lambda), \end{aligned} \quad (6)$$

where $\Lambda = \text{diag}(\lambda)$. Since the first and second order derivatives of this Lagrange dual form can be easily obtained, problem (5) can then be optimized by using Newton's method or conjugate gradient (Chong and Żak 2008).

Problem (4) is a SVM formulation (Scholkopf and Smola 2002) and can be solved either in the primal form or the dual form. Similar to (Joachims 2006), we apply the Cutting Plane method (Kelley 1960) to accelerate the training process. The basic motivation for the cutting plane algorithm is to find a small set of most meaningful examples iteratively for training, rather than all of them. This method has

Algorithm: Transfer-LSI

Input:

1. Source Domain Examples: $\{(\mathbf{z}_1, \mathbf{y}_1), \dots, (\mathbf{z}_m, \mathbf{y}_m)\}$
2. Target Domain Examples: $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$
3. Parameters: C_1, C_2, β , as in Eq.(1), the dimension of the hidden space s , and precision $\epsilon = 0.01$.

Output: the labels of the examples $\{\mathbf{y}_1^l, \mathbf{y}_2^l, \dots, \mathbf{y}_n^l\}$

Latent Semantic Space Inference:

1. Randomly initialize \mathbf{A} .
- repeat**
2. $\mathbf{A}_{old} = \mathbf{A}$.
3. Compute Φ by solving problem (3).
- for** $i = 1 : l$
4. Compute \mathbf{w}_i and b_i by solving problem (4).
- end for**

5. Obtain \mathbf{A} by solving problem (7).

until $\|\mathbf{A} - \mathbf{A}_{old}\|^2 \leq \epsilon$

Learning on the Target Domain:

- for** $i = 1 : n$
6. $\mathbf{a}_i = (\tilde{\Phi}_i^T \tilde{\Phi}_i + \beta \mathbf{I})^{-1} \tilde{\Phi}_i^T \tilde{\mathbf{x}}_i$
- end for**

Output: $\mathbf{y}_{ij}^l = 1$ if $\mathbf{w}_j^T \mathbf{a}_i + b_j \geq 0$, and $\mathbf{y}_{ij}^l = 0$ otherwise

Table 3: Description of Transfer LSI Algorithm

already been shown to be more effective than the traditional approaches to solve the SVM formulation.

2. Fixing \mathbf{W} , \mathbf{b} , and Φ , we seek the solution of \mathbf{A} . The optimization problem for solving each \mathbf{a}_i is:

$$\begin{aligned} \min_{\mathbf{a}_i, \xi_j} \quad & C_1 (\|\mathbf{z}_i - \Phi \mathbf{a}_i\|_2^2 + \beta \|\mathbf{a}_i\|_2^2) + C_2 \sum_{j=1}^l \xi_j \\ \text{s.t.} \quad & \forall j \in \{1, \dots, l\} \\ & \mathbf{y}_{ij} (\mathbf{w}_j^T \mathbf{a}_i + b_j) \geq 1 - \xi_j, \quad \xi_j \geq 0, \end{aligned} \quad (7)$$

This is a standard quadratic programming problem (Boyd and Vandenberghe 2004). If l is much smaller than the number of basis in Φ , we can solve it from its dual problem (Boyd and Vandenberghe 2004). By doing so, the number of variables that need to be optimized can be greatly decreased.

Target Domain Reconstruction

In Transfer-LSI, the examples in the target domain are unlabeled and the feature vectors are extremely sparse. By reconstructing the examples in the target domain in the latent space that we have learned previously in the source domain, the ‘‘lost’’ part of these examples can be recovered. In particular, the new activations for \mathbf{x}_i can be computed as follows:

$$c(\mathbf{x}_i) = \arg \min_{\mathbf{a}} \|\tilde{\mathbf{x}}_i - \tilde{\Phi}_i \mathbf{a}\|_2^2 + \beta \|\mathbf{a}\|_2^2, \quad (8)$$

where $\tilde{\mathbf{x}}_i$ is the non-zero part for \mathbf{x}_i and $\tilde{\Phi}_i$ represents the corresponding part of Φ for the non-zero features of \mathbf{x}_i . This is a regularized least square problem, with the optimal solution: $c(\mathbf{x}_i) = (\tilde{\Phi}_i^T \tilde{\Phi}_i + \beta \mathbf{I})^{-1} \tilde{\Phi}_i^T \tilde{\mathbf{x}}_i$. It can be solved efficiently by employing Woodbury inversion (Higham 2002), since the rank of $\tilde{\Phi}_i^T \tilde{\Phi}_i$ is very low. The reason why only the non-zero part of \mathbf{x}_i is used for reconstruction is that the example features in the target domain are too sparse. In this

case, it is highly possible that a lot of the zero features of \mathbf{x}_i are missing features and if they are used for feature reconstruction, the precision cannot be guaranteed. The current formulation approximately expresses \mathbf{x}_i as a linear combination of $\tilde{\Phi}_i$ and this new representation $c(\mathbf{x}_i)$ now serves as the new representation of \mathbf{x}_i . These newly represented examples can be used for clustering, or classified using the classifier \mathbf{W} and \mathbf{b} obtained in the previous step. The complete algorithm for classification is described in Table 3.

Computation Complexity

The proposed method is an iterative method, and can be solved efficiently using some sophisticated convex optimization algorithms. Suppose we have a fixed number of iterations for the latent semantic space inference in Table 3, the time complexity for the algorithm is then dominated by that from step 3 to step 5. For Step 3, as claimed in (Lee et al. 2007), we can solve the dual problem efficiently using Newton’s method or conjugate gradient. Step 4 is a SVM training step, and can be solved efficiently by using the Cutting Plane method (Joachims 2006). For each cutting plane iteration, the time complexity is $O(p_s \times m)$ (p_s is the average number of nonzero features for each example in the source domain) and the total number of iterations for the cutting plane will not exceed a fixed value, as stated in the Lemma 2 of (Joachims 2006). Step 8 is a standard quadratic programming problem. If we solve it through the dual form, the time complexity for solving the dual would be $O(l^2)$. Since there are in total m source domain examples, the complexity for this step should be $O(ml^2)$. For the target domain reconstruction, the time complexity would be $O(n \times s^2)$.

Experiments

We conduct experiments on three datasets, i.e., Synthetic dataset, Reuters-21578, and the Twitter dataset. The datasets are described in detail below and in Table 4.

Dataset

Synthetic Dataset: A synthetic dataset is generated to demonstrate the ability of the proposed method to learn from examples with extremely sparse features through another set of related examples. In the source domain, each example is randomly assigned three binary labels. Based on these three randomly generated labels, we generate a 9-dimensional feature vector from a multivariate Gaussian Mixture Model. In the target domain, the same method is used to generate the labels as well as the feature vectors. But, different from the source domain, for each generated example, approximately 70 percents of its features would be set to zero to get the sparse representations. In this way, 2000 source domain, as well as 473¹ target domain examples are generated.

¹Originally, 500 target domain examples were generated. But when we randomly set some features to zero, the features of some examples become all zero. After removing these examples, we get 473 target domain examples.

	#Dim	# Labels	Source Domain		Target Domain	
			# Inst	Sparsity	# Inst	Sparsity
Synthetic Dataset	9	3	2000	1	473	0.310
Reuters-21578	1029	57	10376	0.043	10305	0.003
Twitter	1416	566	12091	0.161	9995	0.004

Table 4: Datasets Description experiments.

Reuters-21578: This dataset contains documents collected from the Reuters newswire in 1987². There are in total 135 categories associated with 21578 documents in this dataset. We remove documents without any titles and labels. This dataset is further divided into two different domains. One contains the content part of each document, while the other one contains the title part. The content part is used as the source domain examples, and the title part is considered as the target domain examples, since the number of words in the title part is highly limited. Some words in the titles are further randomly removed. The tf-idf (normalized term frequency and log inverse document frequency) (Manning, Raghavan, and Schtze 2008) features of the most frequently appeared words in the source domain are selected for each document³. The same vocabulary and word statistics are used for extracting feature vectors in the target domain. Furthermore, we remove the source-domain examples with zero features and labels appearing less than 20 times, as well as the target-domain examples with only zero features

Twitter: We obtained this dataset from an IT company, and are interested in analyzing the tweets around this company. The source domain dataset includes 12,091 webpages that were tagged by the employees of this company using an internal social bookmarking tool. We then searched for the name of this company in twitter and collected around 9,995 tweets over a period of time. These tweets are used as the target domain dataset. For both the source domain and target domain examples, the tf-idf features are extracted in exactly the same way as we did on Reuters-21578 dataset.

Baseline Methods

For the proposed method, there are three parameters that need to be tuned, i.e., C_1 , C_2 and β , as in Eq.(1). The dimensionality of the hidden space is fixed to be 500 for Reuters-21578 and Twitter dataset, and 20 for the synthetic dataset. The proposed method is compared with three other baseline algorithms, i.e., Support Vector Machine (SVM) (Scholkopf and Smola 2002), Large Margin Transductive Transfer Learning (LMTTL) (Quanz and Huan 2009), and Self Taught Learning (STL). Their parameters are all set through five fold cross validation.

Evaluation Results

To compare the performance of different methods, the average G-mean value (Kubat, Holte, and Matwin 1998) is used here. G-mean is a commonly used criteria in tasks, whose datasets are imbalanced, and is defined by $Gmean =$

²<http://www.daviddlewis.com/resources/testcollections/reuters21578/>

³We removed the stop words, and used porter as the stemmer.

$\sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}}$, where TN , TP , FP , FN represent the numbers of True Negative, True Positive, False Positive, False Negative examples, respectively. The experiments are conducted in a multi-label classification setting, and therefore the average G-mean values across the different labels are reported. For comparison, the average true positive rate (TP rate), and average true negative rate (TN rate) across the multiple labels are also reported.

We report the classification results on the Synthetic dataset, Reuters-21578 in Table 5. We have also compared the performance of these different algorithms on these two datasets with varying number of source domain examples in Fig. 1. In particular, for each experiment, we vary the number of source domain examples by randomly sampling from the whole source domain corpus. The average results of 10 independent runs are reported.

It can be seen from Table 5 that the proposed method achieves the best performance in most cases. We believe this is because the proposed method finds an effective feature transformation that can serve as the latent semantic space for both the source domain and target domain examples, as well as maximizing the performance of classifiers. Furthermore, it recovers the target domain examples by using the learned latent semantic space in an effective way.

Another interesting observation we can see from the results is that the performance of SVM is very competitive, compared to LMTTL and STL, even though it is not a transfer learning algorithm. It means that traditional transfer learning methods do not work well on these datasets due to the sparsity issue. For example, the basic motivation of LMTTL is to find a feature transformation that minimizes the distribution difference between two domains and maximizes the performance of the classifier simultaneously. It works best in cases when the effects of missing features can be omitted. However, in our experiments, the overall distribution of the target domain is significantly different from that of the source domain due to the missing features, which obviously violates the assumption in most transfer learning algorithms. So, we need to recover the target domain examples before classifying them. Furthermore, we can see from Fig.1 that in the synthetic dataset, the performance of LMTTL even decreases a little bit when the number of source domain examples is increased, due to the significant distribution difference between source and target domains.

STL tries to identify a latent space from the examples in the source domain through sparse coding. We note that their method does not work well in our setting. We believe this is mainly due to two reasons: (1) STL does not use the labeled information in the source domain; (2) It does not consider the problem of sparsity in the target domain.

As shown in Fig. 1, the performances of all the methods do not change significantly with the increase of examples in the source domain on the synthetic dataset while it does on Reuters-21578. This is because on the synthetic dataset, the missing features are randomly selected by using the same scheme. So, with more examples, the performance may not be greatly improved. But on Reuters-21578, the statistics for the missing features are different for different examples. So,

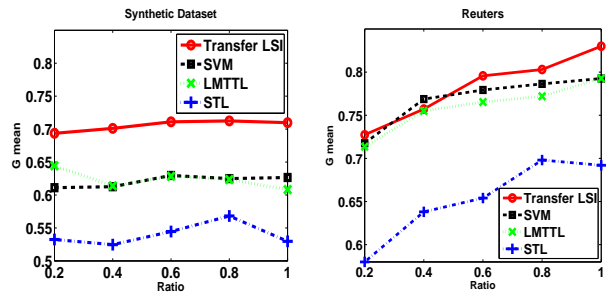


Figure 1: G-mean results on the Synthetic Dataset and Reuters-21578.

training with more examples indicates a better understanding of the dataset, and bring in an improvement on the classification performance.

	Synthetic Dataset			Reuters-21578		
	G-Mean	TP rate	TN rate	G-Mean	TP rate	TN rate
Transfer-LSI	0.710	0.862	0.615	0.830	0.861	0.819
SVM	0.627	0.869	0.540	0.793	0.833	0.767
LMTTL	0.608	0.805	0.523	0.783	0.842	0.749
STL	0.530	0.561	0.464	0.692	0.774	0.695

Table 5: Performance Comparison of four methods

	Top 1	Top 2	Top 5	Top all
Precision	84.25%	80.37%	74.35%	65.89%

Table 6: Evaluation of Transfer LSI on Twitter Dataset: Precision of the predicted tags. In particular, the precisions for the top 1, 2, 5, and all of the returned tags are reported

Results on Twitter

The latent semantic space learned by Transfer-LSI can be applied to both the supervised applications and unsupervised applications. In this section, we report the experimental results of these two types of applications on the Twitter dataset respectively by using Transfer-LSI.

Supervised Application In the supervised application, we aim to predict possible tags of the tweets by using the learned latent semantic space and classifiers. It is fair to assume that both the source domain (i.e., the dogear) and the target domain (twitter) share the same label space, since they are all related to the specific company. Therefore, we classify tweets into the same tag space as the source domain. However, as can be seen from Table 4, there are in total 566 tags/labels. It is unrealistic to evaluate them manually by reading the content of the tweet and selecting the relevant tags from all of the 566 ones. So, instead, we record the results of the proposed method for these tweets, and ask several volunteers to evaluate these top (with the highest posterior probability) 10 returned results. The evaluation results are reported in Table 6. The precisions for the top 1 and top 2 tags are very high (above 84% and 80% respectively), which clearly demonstrate the performance of the proposed

Tweets	Tags
IBM LOTUS SAMETIME GATEWAY ALERT: new inbound talkz.l.google.com IP address 74.125.154.86	guidance,access,technic,workplace,diagnostic
General Other Specialty Sales at IBM (Washington, DC): Data Management sales specialist	computer,software, sell,Rational
Graphic Design Job - Student/Intern - IBM - Somers, CT: 2 year experience in Graphic Design English: Fluent... r...	computer, available, capability, ibmintern
IBM Plans online apos;serious apos; SimCity-style urban-sprawl sim, CityOne (InformationWeek):	computer, todo, toolbar,diagram,recommend,time.brand
IBM Perspective on Cloud Computing	assess, brief,introduction, usage.operation
Hiring a Systems Administrator at IBM (Washington, DC) #jobs #shjobs	find, expertis, firewall, site,
TheBPMNetwork: #BPM #Careers Lombardi Business Process Management (BPM) Consultant -IBM-United States	technic, webservice,function,websphere
Information Developer at IBM (Washington, DC): THIS IS AN INTERNSHIP POSITION AND ALL	technic,computer,enable, software,US
IBM: Account Manager (Los Angeles, CA)	chat, workplace, role, account
IBM to participate in the Transportation Security Forum next week (May 3-4). Visit us in booth #A07!	technic, workshop, export, function, time
IBM presents, System X3650 M2 Express-7947 ISG,worth Rs10,000 et Rs5,000/- off	enable, design,computer, setup
Stupid IBM software update tool!! At least warn me before rebooting my laptop!!!!!!	essential.inform, ibm, lotus,setup,

Table 7: Supervised Application for Transfer-LSI: Example tweets and their predicted tags by Transfer-LSI.

Cluster Interpretation	Cluster Summary
Software Development	'practition' 'solution' 'hub' 'dashboard' 'ISO' 'Agile' 'firewall' 'user' 'configure'
IBM Software Conferences	'conference' 'technic' 'Rational' 'IM' 'chart' 'Lotosphere' 'software' 'Infosphere' 'coe' (Center of Excellence)
Lotus Products	'webconference' 'assist' 'Symphony' 'pilot' 'Lotus Notes' 'Quickr' 'skill'
Instant Messaging Tools	'workplace' 'IM' 'role' 'chat' 'audio'
Software Sales	'Domino' 'UK' 'National' 'sell' 'SUT' (Sametime Unified Telephony)
Business Engagements	'COP' (Communities of Practice) 'UI' 'engagement' 'IM' 'Business' 'practition'
Webconference Software	'reader' 'quick' 'computer' 'webconference' 'record' 'free'
Social Software	'communication' 'socialsoftware' 'favorite' 'aggregate' 'mobile' 'w3ki'

Table 8: Unsupervised Application for Transfer-LSI.

method. We also report the tag assignments of some sample tweets in Table 7. Most of the tags are very reasonable.

Unsupervised Application Another interesting topic for Twitter mining is to discover the hidden topics underlying tweets. Since we have recovered the tweets on the shared latent semantic space, we can conduct clustering based on that and give a summary for each of the clusters. In particular, after getting the new representations for the 9995 tweets based on the latent semantic space by using Transfer-LSI, they are further grouped into 30 clusters by using k-means (Duda, Hart, and Stork 2001). The clusters and corresponding tags for 8 clusters are reported in Table 8. It can be observed that the clustering is useful in identifying various conversations around this IT company. For example, software development is an important theme, with discussions centering around practioner, solutioning and Agile development methodology.

Conclusion

In this paper, we propose a novel transfer learning approach, namely, Transfer Latent Semantic Learning (Transfer-LSI). The proposed method utilizes a large number of related tagged documents (source domain) to improve the learning on examples with highly sparse features (target domain). In particular, it learns a latent semantic space shared by both the source domain and the target domain by simultaneously minimizing the document reconstruction error and the error in the classification model learned in the latent space on the source domain. Then it reconstructs the target domain examples based on this learned latent space. Experimental results on a synthetic dataset, Reuters-21578, and a Twitter dataset clearly demonstrate the advantages of our approach over other state-of-the-art methods. For future work, we are interested in providing theoretical analysis of the algorithm and exploring effective approaches to obtain related tagged documents for analyzing microblogging data on any topics.

Acknowledgement

We would like to express our sincere thanks to Estepan Meliksetian, and the anonymous reviewers for their valuable comments and suggestions.

References

- Arnold, A.; Nallapati, R.; and Cohen, W. 2008. A comparative study of methods for transductive transfer learning. In *ICDM Workshops*.
- Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge Univ Press.
- Chong, E., and Žak, S. 2008. *An introduction to optimization*. Wiley-Interscience.
- Duda, R.; Hart, P.; and Stork, D. 2001. *Pattern classification*.
- Higham, N. 2002. *Accuracy and stability of numerical algorithms*. Society for Industrial Mathematics.
- Joachims, T. 2006. Training linear SVMs in linear time. In *KDD*.
- Kelley, J. 1960. The cutting plane method for solving convex programs. *Journal of the SIAM*.
- Kubat, M.; Holte, R.; and Matwin, S. 1998. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*.
- Lacoste-Julien, S.; Sha, F.; and Jordan, M. 2008. DiscLDA: Discriminative learning for dimensionality reduction and classification. *NIPS*.
- Lee, H.; Battle, A.; Raina, R.; and Ng, A. 2007. Efficient sparse coding algorithms. *NIPS*.
- Manning, C. D.; Raghavan, P.; and Schtze, H. 2008. *Introduction to Information Retrieval*.
- Pan, S., and Yang, Q. 2009. A survey on transfer learning. *TKDE*.
- Quanz, B., and Huan, J. 2009. Large margin transductive transfer learning. In *CIKM*.
- Raina, R.; Battle, A.; Lee, H.; Packer, B.; and Ng, A. 2007. Self-taught learning: transfer learning from unlabeled data. In *ICML*.
- Scholkopf, B., and Smola, A. 2002. *Learning with kernels*. MIT press Cambridge.