Knowledge Transferring Via Implicit Link Analysis

Xiao Ling, Wenyuan Dai, Gui-Rong Xue, and Yong Yu

Department of Computer Science and Engineering
Shanghai Jiao Tong University
No. 800 Dongchuan Road, Shanghai 200240, China
{shawnling,dwyak,grxue,yyu}@apex.sjtu.edu.cn

Abstract. In this paper, we design a *local* classification algorithm using *implicit link analysis*, considering the situation that the labeled and unlabeled data are drawn from two different albeit related domains. In contrast to many *global* classifiers, e.g. Support Vector Machines, our local classifier only takes into account the neighborhood information around unlabeled data points, and is hardly based on the global distribution in the data set. Thus, the local classifier has good abilities to tackle the non-*i.i.d.* classification problem since its generalization will not degrade by the bias w.r.t. each unlabeled data point. We build a local neighborhood by connecting the similar data points. Based on these *implicit links*, the Relaxation Labeling technique is employed. In this work, we theoretically and empirically analyze our algorithm, and show how our algorithm improves the traditional classifiers. It turned out that our algorithm greatly outperforms the state-of-the-art supervised and semi-supervised algorithms when classifying documents across different domains.

1 Introduction

Supervised classification [1,2] requires a large number of labeled data. However, manual labeling is very expensive and time-consuming. Many investigations focus on the situations that the labeled data are scarce, and the unlabeled data are used to enhance the classification performance [3,4]. Actually, most of these researches ignore the fact that there might be quite a lot of existing labels from the similar domains. For example, the Blog documents and Web pages come from different albeit related domains; there are hardly any labeled Blog documents, while there are plenty of labeled Web pages, e.g. the pages in Open Directory Project (ODP). It is quite wasteful not to use these label information. However as a result of the domain difference, their distributions differ due to the different word usage for the documents in the two domains. The non-*i.i.d.* data violate the basic assumption of the traditional classification techniques, and thus the traditional classifiers cannot cope well with the *cross-domain* classification problem. Note that the *cross-domain learning*, which is one simplified case of *transfer learning* [5,6,7,8], transferring knowledge across tasks and domains.

In this paper, we focus on the problem of classifying documents across domains. Recall the Web page and Blog entry example. The training data (the Web pages) are under domain $\mathcal{D}_{in}$ and the test data (the Blog entries) under domain $\mathcal{D}_{out}$ are available. We call $\mathcal{D}_{in}$ *in-domain*, and $\mathcal{D}_{out}$ *out-of-domain*. In addition, it is assumed that $\mathcal{D}_{out}$ and

J.R. Haritsa, R. Kotagiri, and V. Pudi (Eds.): DASFAA 2008, LNCS 4947, pp. 520–528, 2008.

$\mathcal{D}_{in}$ are related and share some common knowledge, which makes knowledge transferring feasible. Our general goal is to classify the test data from $\mathcal{D}_{out}$ accurately by transferring knowledge from the labeled data from $\mathcal{D}_{in}$.

We regard this classification problem as a labeling problem in a graph where both labeled and unlabeled documents are represented as nodes. The edges are built based on the *similarity* between two nodes. Such connections are so-called *Implicit Link*s. Based on the assumption of Markov Random Fields (MRF) that the label of each node is only dependent on its immediate neighbors, we adopt the *Relaxation Labeling* [9] technique to address the labeling problem. Initially, the labels for those unlabeled data are assigned using a global classifier and then these labels are iteratively updated according to the local neighborhood information. Since both labeled and unlabeled documents may exist among the neighbors due to the similarity of domains, the iterative adjustments are in fact implicitly transferring knowledge to the target domain. This is why our local classifier is capable of handling the cross-domain problems.

Some prior works use labeled data from in-domain to solve problems under the target domain. Wu & Dietterich [10] investigated how to exploiting in-domain data in *k-Nearest-Neighbors* and SVM algorithm. Daumé III and Marcu [11] utilized additional in-domain labeled data to train a statistical classifier under the *Conditional Expectation Maximum* framework. Those in-domain data play a role as auxiliary data in tackling the scarcity of out-of-domain training data. In these work, the auxiliary data serve as a supplement to the ordinary training data. In contrast, our work do not need any training examples in the target domain. Note that, it is possible, because the in-domain and out-of-domain data share come common knowledge as we assumed, for the in-domain model to learn from the out-of-domain data.

2 Transferring Knowledge through Relaxation Labeling

2.1 Problem Definition

For conciseness and clarity, we mainly focus on binary classification on the textual data from different domains. Given two document sets $\mathcal{S}_{in}$ and $\mathcal{S}_{out}$ from in-domain $\mathcal{D}_{in}$ and out-of-domain $\mathcal{D}_{out}$ respectively, each element $\mathbf{d}_i$ in two sets is represented by a feature vector. In the binary classification setting, the label set is $\{+1, -1\}$, that is $c(\mathbf{d}_i)$ equals $+1$ (positive) or -1 (negative) where $c(\mathbf{d}_i)$ is $\mathbf{d}_i$'s true label. As assumed in Section 1, $\mathcal{D}_{in}$ and $\mathcal{D}_{out}$ are different albeit related. The objective is to find the hypothesis h which satisfies $h(\mathbf{d}_i) = c(\mathbf{d}_i)$ for as many $\mathbf{d}_i \in \mathcal{S}_{out}$ as possible.

2.2 Local Classifier Using Labeled Neighbors

When only the content information is considered, the most probable class label c_i for each document d_i maximizes $\Pr(c_i|\tau(d_i))$ where $\tau(d_i)$ is the textual information of d_i. However, the fact that the labeled and unlabeled data come from different domains curbs the generalization ability since the model will fit the training data, but will not cope well with the test data.

In order to circumvent this obstacle, the class labels of similar documents are also worthy considering. We build a graph with nodes representing documents and edges by

implicit links. Hereinafter, the terms "document" and "node" are used interchangeably. Each document is connected to its most similar documents. With these links, we prefer the class label which maximizes $\Pr(c_i|\tau(d_i), N_i)$ where N_i is the immediate neighborhood of d_i. This immediate neighborhood assumption characterizes the first-order *Markov Random Field*. In this subsection, it is assumed that the labels of neighbors are all known, although this assumption does not hold in our problem setting. In the next subsection, the model will be extended to cope with neighbors without labels. Applying the Bayes Rule to $\Pr(c_i|\tau(d_i), N_i)$, it is obtained that

$$\Pr(c_i|\tau(d_i), N_i) = \frac{\Pr(\tau(d_i), N_i|c_i) \cdot \Pr(c_i)}{\Pr(N_i, \tau(d_i))}. \tag{1}$$

Assume the content of the document $\tau(d_i)$ has no direct coupling with its neighbors' labels. And $\Pr(N_i, \tau(d_i))$ is regarded as a constant since the task is to classify d_i. Then (1) is spanned into

$$\Pr(c_i|\tau(d_i), N_i) \propto \Pr(\tau(d_i)|c_i) \cdot \Pr(N_i|c_i) \cdot \Pr(c_i). \tag{2}$$

Assuming that given the class label of a node d_i, all its neighbors are independent with each other,

$$\Pr(N_i|c_i) = \prod_{d_j \in N_i} \Pr(d_j|c_i). \tag{3}$$

Combining (2) and (3), we obtain that

$$\begin{aligned} c_i &= \arg\max_{c_i} \Pr(c_i|\tau(d_i), N_i) \\ &= \arg\max_{c_i} \Pr(\tau(d_i)|c_i) \cdot \Pr(c_i) \prod_{d_j \in N_i} \Pr(d_j|c_i). \end{aligned} \tag{4}$$

2.3 Classification with Out-of-Domain Unlabeled Data

As mentioned in the last subsection, the assumption that the labels of all neighbors are known is hardly satisfied. It is to say that all the similar documents of an unlabeled document are labeled, which is rarely possible in the cross-domain setting. To utilize the neighbors without labels, the *Relaxation Labeling* (abbreviated as RL) [12] technique is adopted here. In the RL process, with the initial labels, updates for unlabeled data are carried out iteratively.

Intuitively, the neighborhood of a certain node d is more likely to be given the same label of d. Both the test instances and the training ones are allowed to be the neighbors of the test nodes. The neighbors from the training data partially supervise the labeling while at the same time the test neighbors help not only correctly update labels but also avoid the bias by the constraints of local consistency. The Relaxation Labeling technique here reduces the cross-domain bias because in the iteration it enables the unlabeled data to be classified by themselves. In this view, the Relaxation Labeling updates the labels iteratively and thus gradually transfers knowledge across domains.

With the implicit links in previous subsection, we denote G^K to be all the information known in the graph. In this notation, the most probable label c_i is the one that can maximizes

$$\Pr(c_i|G^K) = \sum_{N_i^U} \Pr(c_i|G^K, N_i^U) \cdot \Pr(N_i^U|G^K) \tag{5}$$

where c_i is the class label corresponding to d_i and N_i^U represents the set of d_i's neighbors still with "unknown" label. The summation is over all possible assignments of N_i^U.

Using the independence assumption of the class label for each d_j among N_i^U,

$$\Pr(N_i^U|G^K) = \prod_{d_j \in N_i^U} \Pr(c_j|G^K)\,. \tag{6}$$

Similarly with previous subsection, the class label of one document is dependent on its local content as well as its similar documents (i.e. its immediate neighbors).

$$\Pr(c_i|G^K, N_i^U) = \Pr(c_i|N_i^K, N_i^U) \tag{7}$$

where N_i^K is the neighborhood with "known" labels. Combining (6) and (7) and manipulating it into an iterative solution, we obtains

$$\Pr(c_i|G^K)^{(r+1)} = \sum_{N_i^U} \left[\prod_{d_j \in N_i^U} \Pr(c_j|G^K)^{(r)} \Pr(c_i|N_i^K, N_i^U)^{(r)} \right] \tag{8}$$

where $\Pr(c_i|N_i^K, N_i^U)$ can be treated as $\Pr(c_i|N_i)$ in the previous subsection where the labels of N_i are all known. The superscript (r) denotes the iteration number.

Since the number of the terms in the summation (8) is exponential to the size of unlabeled documents, the computation is intractable. To reduce the computation expense, we adopted the *hard labeling* method in [13], whose main idea is to use the most probable initial labels of those unknown neighbors to alleviate the consuming computation of summation.

$$\Pr(c_i|G^K) \approx \Pr(c_i, N_i^{U'}|N_i^K) \tag{9}$$

where $N_i^{U'}$ is the neighborhood with the most probable assignment for class labels. This hard labeling is seen as a rough approximation of (8). However, the magnitude of other terms is often small compared to the selected assignment, and therefore the hard labeling method may work well. We also consider the soft version of labeling strategy [13], which selectively takes more terms of the summation (8) into computation. Empirically, it is comparable to the "hard labeling" strategy. The details are omitted due to the space limit. Algorithm 1 gives the outline of our method. After the initializations, the algorithm iterates until the convergence and then it outputs the predicted labels of unlabeled data.

Algorithm 1. Transfer Knowledge by Relaxation Labeling (TKRL)

Input :
labeled and unlabeled data from in-domain and out-of-domain respectively,
the initial labels for unlabeled data via a basic classifier,
parameter k for building the graph.
Output : the final labels for unlabeled data
Initialization:
$oldlabel$ = $null$, $newlabel$ = initial labels,
build the graph with all information including the content of each d_i and each immediate neighborhood N_i, s.t. $|N_i| = k$ for each i.
Iteration:
while $oldlabel \neq newlabel$ **do**
  $oldlabel$ = $newlabel$.
  estimate the prior probability Pr$(c = +1)$ and Pr$(c = -1)$.
  for each d_i **do**
    estimate the conditional probabilities Pr$(c_i = +1|d_i)$, Pr$(c_i = -1|d_i)$
  end for
  for d_i in unlabeled data **do**
    update its label in $newlabel$ according to (9)
  end for
end while
return $newlabel$

3 Experimentation

3.1 Data Sets

To validate our algorithm, we developed a series of cross-domain data sets based on 20 Newsgroups[1], Reuters-21578[2] and SRAA[3]. The basic idea is to utilize the hierarchy of the data sets. The task is defined as classifying top categories. Each top category is split into two disjoint parts with different sub-categories, one for training and the other for test. Therefore the training and test data come from different domains. Take SRAA as an example, which is a Simulated/Real/Aviation/Auto UseNet data set for document classification. For the data set `real vs simulated`, we use the documents in `real-auto` and `sim-auto` as in-domain data, while `real-aviation` and `sim-aviation` as out-of-domain data. Other tasks were generated in a similar way. On these textual data, regular preprocessing was done including tokenization into bag-of-words, converting into low-case words, stop-word removing and stemming. We also carried out feature selection by thresholding Document Frequency [14]. In our experiments, Document Frequency threshold is set to 3.

The data from different domains are certainly under different distributions. To verify our data design, we calculated *Kullback-Leibler Divergence* (K-L Divergence) [15] based on Term Frequency for each data set, which measures distance between

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/
[2] http://www.daviddlewis.com/resources/testcollections/
[3] http://www.cs.umass.edu/ mccallum/data/sraa.tar.gz

Table 1. Description of the data sets for cross-domain text classification, and the error rates of each classifier. "$\mathcal{D}_{in}$–$\mathcal{D}_{out}$" means training on $\mathcal{D}_{in}$ and testing on $\mathcal{D}_{out}$; "$\mathcal{D}_{out}$–CV" means 10-fold cross-validation on $\mathcal{D}_{out}$.

Data Set	Documents			K-L	SVM		TSVM	NBC	TKRL
	$\lvert\mathcal{D}_{in}\rvert$	$\lvert\mathcal{D}_{out}\rvert$	$\lvert\mathcal{W}\rvert$		$\mathcal{D}_{out}$–CV	$\mathcal{D}_{in}$–$\mathcal{D}_{out}$			
real vs simulated	8,000	8,000	14,433	1.161	0.032	0.266	0.130	0.245	**0.126**
auto vs aviation	8,000	8,000	14,433	1.126	0.033	0.228	0.102	0.136	**0.099**
rec vs talk	3,669	3,561	19,412	1.102	0.003	0.233	0.040	0.269	**0.032**
rec vs sci	3,961	3,965	18,152	1.021	0.007	0.212	0.060	0.153	**0.058**
comp vs talk	4,482	3,652	17,918	0.967	0.005	0.103	0.097	0.025	**0.022**
comp vs sci	3,930	4,900	18,379	0.874	0.012	0.317	0.183	0.206	**0.100**
comp vs rec	4,904	3,949	18,903	0.866	0.008	0.165	0.098	0.216	**0.046**
sci vs talk	3,374	3,828	20,057	0.854	0.009	0.226	0.108	0.258	**0.056**
orgs vs places	1,079	1,080	4,415	0.329	0.085	0.454	0.436	0.375	**0.339**
people vs places	1,239	1,210	4,562	0.307	0.113	0.266	0.231	0.217	**0.188**
orgs vs people	1,016	1,046	4,771	0.303	0.106	0.297	0.297	0.282	**0.272**

distributions. More formally, $\mathrm{KL}(\mathcal{D}_1||\mathcal{D}_2) = \sum_i \mathcal{D}_1(i) \log_2 \frac{\mathcal{D}_1(i)}{\mathcal{D}_2(i)}$ where $\mathcal{D}_1$ and $\mathcal{D}_2$ are two distributions. As listed in the fifth column of Table 3.1, the K-L Divergence values of all the data sets are all far larger than zero which means that they come from different distributions. This observation justifies that our design is reasonable. Also, we calculated the error rates using the SVM classifier across the domains ($\mathcal{D}_{in}$–$\mathcal{D}_{out}$) and only within the test set (D_{out}–CV). The relative low error rates in D_{out}–CV prove that the test data are out-of-domain.

3.2 Experimental Results

To evaluate the effectiveness of our method, we compare it to two supervised methods: the SVM and the Naive Bayes classifier (NBC) as well as a semi-supervised method: the TSVM (Transductive SVM) classifier [4] by their error rates. The SVM and TSVM classifiers are implemented by SVM^{light} [4]. The Naive Bayes Classifier is implemented using Laplace Smoothing. Each document is then represented by a feature vector with Term Frequency in our algorithm. When applying SVM or TSVM to these data (mentioned in the next subsection), the tf-idf values are used. Through comparing with traditional supervised classifier, it is seen that the different domains the training and test data come from bring classification much difficulty and hence poor performance. Although the semi-supervised classifier fully utilizes the unlabeled data in the classification process, it still works under the identical-domain assumption.

Our method (named TKRL) aims at handling the cross-domain problem, which achieves high performance in this cross-domain data setting. In the implementation, we use the Naive Bayes classifier to give the initial labels and adopt cosine measure for building the graph. In Table 3.1, we see that semi-supervised algorithm TSVM (8th column) always outperforms the supervised algorithm SVM (7th column) and NBC (9th column) almost all the time. It is because taking unlabeled data into account is in some

[4] http://svmlight.joachims.org

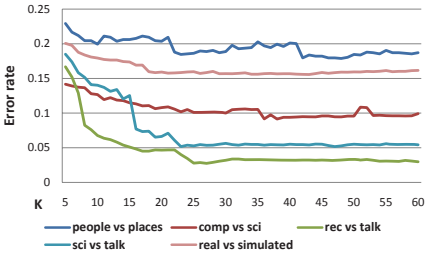

Fig. 1. The error rate curve against parameter k

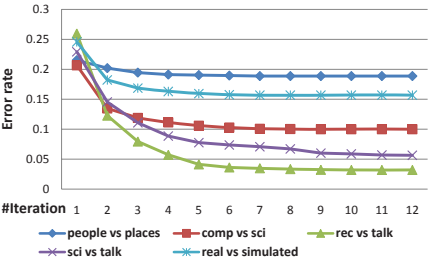

Fig. 2. The convergence curve of five tasks

sense partially transferring supervisory knowledge into the target domain. However the transferring is not complete. On the other hand, it is noticed that NBC performs better than SVM. We believe that NBC is less influenced by the domain difference than SVM due to its simple independence assumption. Employing implicit link analysis, our method aims at handling data under different domains and in fact TKRL achieves the lowest error rates through all eleven tasks. However, the performance of certain data sets are still unsatisfactory. It is mainly attributed to the noise in the data. In the three Reuters-21758 tasks, the test error by SVM is not satisfactory yet. It is mainly because of the data noise and thus less common knowledge between domains. Note that our algorithm achieves improvements on the classical classifiers.

Parameter Sensitivity. Only one parameter k exists in our algorithm, which limits the size of immediate neighborhood. We enumerate the value of k ranging from 5 to 60 to evaluate its influence on performance. Figure 1 displays the error rate curve on the five representative tasks. It is observed that our algorithm is not very sensitive to k when k is greater than 30 since the rest of the curves are quite stable. Empirically, we set $k = 30$ to get better performance.

Convergence. In Fig. 2, we plot the error rate along with each iteration step on five tasks. Experimentally, our algorithm converges after several iterations. Generally, the

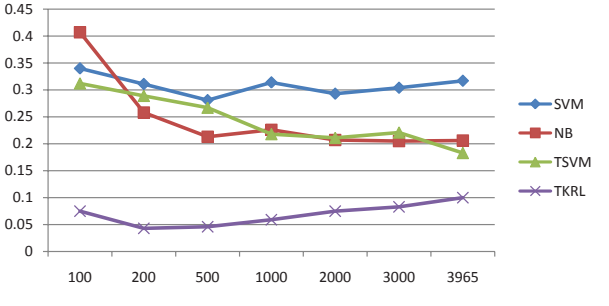

Fig. 3. The error rate curve against the size of training data

iteration process needs around 9 steps on average. From Fig. 2, we observe that the error rate decreased by a large amount in the first several iterations.

Size of Training Examples. We also investigate the influence by the size of training examples. A portion of examples in `comp vs sci` are randomly chosen for training, from 100 examples to all. From Fig. 3, we observed that TKRL reaches the lowest error rate at the size of 200 training examples. It is because if the training data are fewer, the information from labeled data will be too scarce; on the contrary, if the data from in-domain are more than enough, the in-domain knowledge will impact and deteriorate the out-of-domain classification performance.

4 Conclusion and Future Work

In this paper, we design a method for the cross-domain classification problem where only labeled data from in-domain are available for predicting the class labels of unlabeled data from out-of-domain. Our local classifier labeled the test data only considering the neighborhood information. We leverage implicit link analysis for this cross-domain classification. Experimental evaluations reveal that our method is very effective on handling the cross-domain problems. There are several directions for future extensions. We wish to test on another kind of data, such as images. It is also interesting to find an online way of classification, that is the test data are incrementing.

Acknowledgements. All authors are supported by a grant from National Natural Science Foundation of China (NO. 60473122). We thank the anonymous reviewers for their great helpful comments.

References

1. Lewis, D.D.: Representation and learning in information retrieval. PhD thesis, Amherst, MA, USA (1992)
2. Boser, B.E., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory (1992)
3. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin–Madison (2006)
4. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of Sixteenth International Conference on Machine Learning (1999)
5. Schmidhuber, J.: On learning how to learn learning strategies. Technical Report FKI-198-94, Fakultat fur Informatik (1994)
6. Thrun, S., Mitchell, T.: Learning One More Thing. IJCAI, 1217–1223 (1995)
7. Caruana, R.: Multitask Learning. Machine Learning 28(1), 41–75 (1997)
8. Ben-David, S., Schuller, R.: Exploiting task relatedness for multiple task learning. In: Proc. of the Sixteenth Annual Conference on Learning Theory COLT 2003 (2003)
9. Pelkowitz, L.: A continuous relaxation labeling algorithm for markov random fields. IEEE Transactions on Systems, Man and Cybernetics 20(3), 709–715 (1990)
10. Wu, P., Dietterich, T.: Improving SVM accuracy by training on auxiliary data sources. In: Proceedings of the Twenty-First International Conference on Machine Learning, pp. 871–878

11. D.I.H., Marcu, D.: Domain Adaptation for Statistical Classifiers. Journal of Artificial Intelligence Research 1, 1–15 (1993)
12. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. In: SIGMOD, pp. 307–318 (1998)
13. Angelova, R., Weikum, G.: Graph-based text classification: learn from your neighbors. In: SIGIR, pp. 485–492 (2006)
14. Yang, Y., Pedersen, J.: A comparative study on feature selection in text categorization. In: Proceedings of the Fourteenth International Conference on Machine Learning (1997)
15. Kullback, S., Leibler, R.: On Information and Sufficiency. The Annals of Mathematical Statistics 22(1), 79–86 (1951)