BAYESIAN MULTI-SOURCE DOMAIN ADAPTATION

SHI-LIANG SUN, HONG-LEI SHI

Department of Computer Science and Technology, East China Normal University
500 Dongchuan Road, Shanghai 200241, P. R. China
E-MAIL: slsun@cs.ecnu.edu.cn, lhshi12@gmail.com

Abstract:

This paper presents a new multi-source domain adaptation framework based on the Bayesian learning principle (BayesMSDA), in which one target domain and more than one source domains are used. In this framework, the label of a target data point is determined according to its posterior, which is calculated using the Bayesian formula. To fulfill this framework, a novel prior of the target domain based on Laplacian matrix and a new likelihood dynamically obtained using the k-nearest neighbors of a data point are defined. We focus on the situation that there are no labeled data obtained from the target domain while there are large numbers of labeled data from source domains. Experiments on synthetic data and real-world data illustrate that our framework has a good performance.

Keywords:

Bayesian framework; multi-source domain adaptation; Laplacian matrix

1. Introduction

Most theoretical models in machine learning, such as probably approximately correct models (PAC models), assume that models are trained and tested using data drawn from certain fixed distributions. Uniform convergence theory guarantees that a model's empirical training error is close to its true error under such assumptions. However, in practice assumptions that data for training and testing come from the same distribution do not hold, because these two types of data usually come from different distributions, or domains. In these cases there is no hope for good generalization. We wish to learn a model in one or more source domains (i.e. domains from which the training data come), and then apply it to a different target domain (i.e. domain from which the test data come). This kind of learning model is called "$domain\ adaptation$" models [1], [2]. We confront this problem in many fields, such as sentimental analysis [3], [4], [5], natural language processing [6], computer vision, etc. Often in these cases, source domains offer large numbers of labeled data for learning, while target domains may have no labeled data available. In this paper, we concentrate on the situation that there are no labeled data in the target domain, and there is only one target domain. The task is to combine the labeled source data and unlabeled target data to classify the target data as correctly as possible.

The problem of multi-source domain adaptation considered in this paper has been researched by many researchers [7], [8], [9], [10]. Crammer et al. [7] considered a problem about learning an accuracy model via the nearby data points from more than one source domains. They gave a general algorithm as follows: by using samples from different source domains to estimate the divergence among these sources, the algorithm determines which samples from each source should be selected to train the model. Thus a corresponding subset that best suits the target task was chosen from each source. On a binary classification task the algorithm was demonstrated to be effective. Tu and Sun [8] gave an emsemble learning framework for domain adaptation. They presented a novel ensemble-based method which dynamically assigns weights to different test examples by using the so-called friendly classifiers. The model gave the most favorable weights to different examples. Mansour et al. [9] presented a theoretical analysis of domain adaptation learning with multiple sources. They gave a combination of the source hypotheses weighted according to the source distributions. In practice they showed that for any fixed target function, there existed a distribution weighted combining rule that has a loss at most ϵ.

An interesting issue to consider in multi-source domain adaptation is that what we should do if we do not know in advance which domain performs best. On one hand, we want to use the most suitable source to solve the target task. On the other hand, we do not know which one to choose. This paper gives a tradeoff for this problem. In this paper, we present a multi-source domain adaptation framework based on the Bayesian learning principle (BayesMSDA). Under the Bayesian framework, the determination of classification is based on the posterior probabilities. These posteriors are proportional to the product of the priors and the likelihoods. We define in BayesMSDA framework a novel prior using the Laplacian matrix [11], [12], and a novel likelihood based on the mean Euclidean distance of k-nearset points.

The remainder of this paper is organized as follows. The new framework and its implementation for multi-source domain adaptation are introduced in detail in Section 2. In Section 3, two experiments on synthetic and real-world data are accomplished to illustrate the effectiveness of our framework. In Section 4, conclusions and the future work are given.

2. The proposed framework

The proposed framework for multi-source domain adaptation (BayesMSDA) is based on the Bayesian learning principle: the probability of which class a target example belongs to is proportional to the product of prior and likelihood assigned to this example. For multi-source issues, the core problem is how to combine these sources effectively to solve the target task. The novel framework is described as follows.

Given M source domains $S_i, i = 1, 2, \dots M$, and one target domain T. The task is to label the data in T, using the unlabeled data in T and the large numbers of labeled data in S_i. Assume that we can get M classifiers c_i, based on the M source domains S_i. Then we define the prior, which measures the fitness between the source domain and the target domain, and the likelihood of each target data, which represents the probability of the target data occuring in the source. Applying the Bayesian learning principle to get a posterior for classification, we can use these posteriors to weight the M classifiers c_i to get a final label for the target data.

The framework we present with self-defined prior and likehood is applicable to the situation that the data are unlabeled in the target domain, which is compared with the majority voting algorithm.

2.1. Prior

Consider a weighted undirected graph $G = (V, E)$, with the data set $V = (x_1, x_2, \dots x_n), E = (e_1, e_2, \dots e_l)$. Assume that G is a connected graph (if not, the process followed can be used on each connected component). Let $Y = (y_1, y_2 \dots y_n)$ be the image of the data set under certain mapping rules. The problem now is how to make the images y_i and y_j as close as possible when the data points x_i and x_j are close. A reasonable criterion to guarantee this is to minimize the objective function

$$G = \sum_i^j (y_i - y_j)^2 W_{ij} \tag{1}$$

under appropriate constrains, where $W_{ij} = e^{-\frac{\|x_i - x_j\|}{T}}$ when x_i and x_j are neighbors, and zero otherwise. Equation (1) means that there is a heavy penalty when the images of neighborhood points x_i and x_j are far away from each other. Minimizing it attempts to make sure that y_i and y_j are close if x_i and x_j are close. This property can be used in binary classification problems effectively.

The prior gives a measurement of fitness when a source classifier is applied on the target task: for a sample x in the target domain, it should be independent with x. In this paper, we construct the prior with the Laplacian matrix [11], [12] of the target domain. We consider the issue that data in the target domain which are all unlabeled are used to quantify the prior. For any Y, the objective function becomes

$$\begin{aligned} G &= \sum_i^j (y_i - y_j)^2 W_{ij} \\ &= \sum_i^j \left({y_i}^2 + {y_j}^2 - 2y_i y_j\right) W_{ij} \\ &= \sum_i {y_i}^2 D_{ii} + \sum_j {y_j}^2 D_{jj} - 2\sum_i^j y_i y_j W_{ij} \\ &= 2Y^T LY \end{aligned}$$

where $L = D - W$ is the Laplacian matrix. Notice that W_{ij} is symmetric and $D_{ii} = \sum_j W_{ij}$ is a diagonal matrix. D provides a natural measure on the vertices of the graph G. The bigger the value D_{ii} (corresponding to the ith vertex) is, the more important the vertex is.

Given n points $x_i \in R^d, i = 1, 2, 3, \dots n$. We construct such an undirected graph $G = (V, E)$, with the neighbors of x_i are its k-nearest neighbors. The steps of generating a Laplacian matrix of the target data set are as follows.

Step 1: (calculating the adjacency matrix A) if x_i and x_j are k-nearest neighbors, let A_{ij}=1 as well as A_{ji}=1, otherwise A_{ij}=0 and A_{ji}=0.

Step 2: (calculating the weight matrix W) one of the variations is the heat kernel:

$$W_{ij} = e^{-\frac{\|x_i - x_j\|}{T}} \tag{2}$$

where $T \in R$.

Step 3: (calculating the Laplacian matrix L) let $D_{ii} = \sum\limits_{j} W_{ij}$, then $L = D - W$ is the Laplacian matrix, which is a symmetric, positive semidefinite matrix.

Once we have a Laplacian matrix, the prior is defined as

$$prior^m = \frac{1}{\sum\limits_{i}^{j} \left(y_i^m - y_j^m\right)^2 W_{ij}} = \frac{1}{2(Y^m)^T L Y^m} \qquad (3)$$

where Y^m is the output of the mth source classifier. In multi-source domain cases, the different priors of each source show the fitness between each source domain and the target domain. The bigger the prior is, the better the corresponding source classifier is.

2.2. Likelihood

The likelihood we define here represents the probability of the instance from the target domain occuring in the source domain, which can also refer to the similarity between the target domain and the source domain. The higher the probability described above is, the better the source classifier is. In this paper we use the mean Euclidean distance of the K-nearest neighbors of instance x (which is from the target domain, and these K-nearest neighbors are from the source domain) to measure this likelihood. For each instance x_i in the target domain, the likelihoods in the different source domains are different, which give a dynamic classifying rule. The likelihood that the target data point x_i occurs in the source domain S_m is defined as

$$Like_i^m = \frac{K}{\sum \parallel x_i - x_j^m \parallel} \qquad (4)$$

where x_j^m, which comes form the mth source, is among the K-nearest neighbors of x_i .

According to the Bayesian learning principle, the posterior is proportional to the product of the prior and the likelihood:

$$post_i^m \propto prior_i^m \times Like_i^m \qquad (5)$$

where $post_i^m$ is the posterior of x_i, based on the mth source domain, $prior_i^m$ is the prior of x_i based on the mth source domain, $Like_i^m$ is the likelihood of x_i based on the mth source domain. The posteriors obtained here are used to weight the source classifiers.

3. Experiments

In this section, we evaluate the proposed framework by experiments on both synthetic and real-world data sets. Each dataset has four domains. In the experiments, every domain is treated as the target domain in turn while the other three as source domains.

We use support vector machines (SVMs) [13], [14] for training and testing. For textual classification SVMs have been found to perform better than other classification methods [13], especially for the sentiment analysis [3]. The kernel we use is RBF kernel. The parameters of SVMs are selected using cross validation for each domain.

3.1. Synthetic data

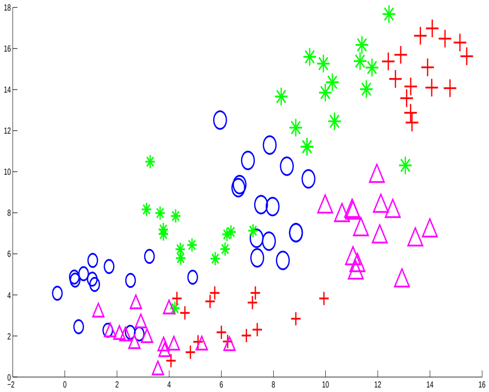

Figure 1. Examples of synthetic data: $\circ$, $\star$, $+$, $\triangle$ stand for a, b, c, d, respectively. The smaller ones (i.e. the bottom-left portion of each domain in the figure) are “positive”, and the larger ones (i.e. the top-right portion of each domain in the figure) are “negative”. Each domain is treated as the “*target*” in turn, while the other three as the “*sources*”

The synthetic dataset consists of four different domains, each of which is sampled from Gaussian distributions with different covariances and means. Figure 1 shows 30 randomly selected data points from each domain, and the different symbols stand for different domains, says, “$\circ$” for a, “$\star$” for b, “$+$” for c, and “$\triangle$” for d. The smaller ones (i.e. the bottom-left portion of each domain in Figure 1) are labeled as “positive”, while bigger

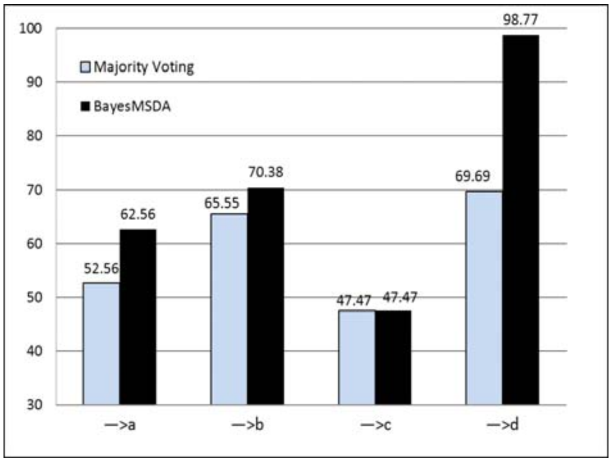

Figure 2. Classification accuracies (%) on synthetic data

ones (i.e. the top-right portion of each domain in the Figure 1) "negative". The base classifiers are trained using SVMs with RBF kernel. Figure 2 illustrates the classification accuracies on these domains using BayesMSDA and the majority voting algorithm. Each domain is treated as the "$target$" in turn, while the other three as the "$sources$".

It is shown in Figure 2 that on three domains (a, b, d), BayesMSDA method outperforms the majority voting method, while on the third one (c) both are equal. Significantly, accuracy of BayesMSDA is far higher than that of the voting method on the fourth dataset (98.77% VS 69.69%). As four datasets are randomly obtained, the results give us a confidence on the effectiveness of the proposed framework. In the next subsection, we apply BayesMSDA on real-world data.

But on the other hand, as we can see, on the third dataset, two results are equal. This phenomenon is acceptable since no such an algorithm can fit the whole situations.

3.2. Real data

Given a piece of text, sentiment classification is a task to determine whether the sentiment expressed by the text is positive or negative. This problem has extended to many new domains, such as stock message boards, congressional floor debates, and blog reviews. Research results have been used to gauge market reaction and summarize opinion from web pages, discussion boards, and blogs.

We use the publicly available data sets [1] from Amazon website in our experiments [1], where there are many reviews for several different types of products. We select four domains: books, DVDs, kitchen & housewares, and electronics (B, D, K, E for short, respectivelly). Each review consists of a rating (1-5 stars), a title, review text, and some other information which are ignored. We make it a binary classification task by binning reviews with 4-5 stars as $“positive''$ and 1-2 stars as $“negative''$, while reviews with 3 stars are discarded.

TABLE 1. Accuracies of One-to-One Classifiers. (T for Target Domain and S for Source Domain)

S T	B(%)	D(%)	E(%)	K(%)
B	-	**79.8**	68.7	65.5
D	**78.8**	-	69.6	69.3
E	65	69.2	-	**78.9**
K	63.6	70.6	**81.4**	-

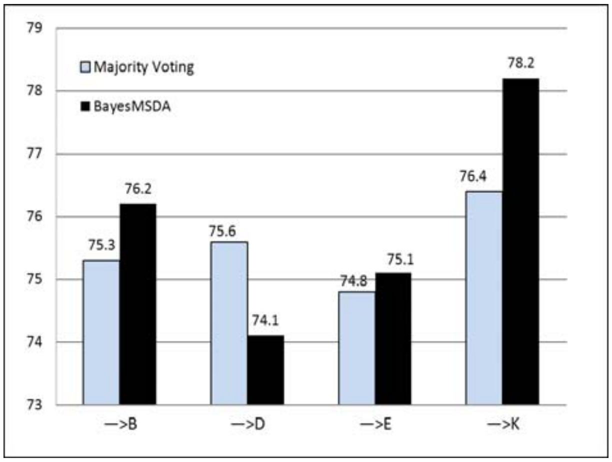

Figure 3. Classification accuracies (%) on real-world data

As vocabularies of reviews for different products vary vastly, classifiers trained on one domain may not fit a different domain because some important lexical information may be missed. This phenomenon motivates us to combine more than one source domains to avoid the shortcomings.

Every domain contains 1000 positive reviews (P) and 1000 negative reviews (N). In our experiments we randomly choose 1000 out of these 2000 instances in each domain for computational convenience. So we have 1000 $\times$ 4 instances in all. Each instance is represented as a sparse feature vector. The feature sets consist of the unigram that occur 5 to 1000 times in all the

[1] http://www.cs.jhu.edu/∼mdredze/

reviews.

At the very beginning, one-to-one linear classifiers are firstly trained without adaptation. These classifiers are regarded as baselines. The baselines are trained using SVMs with RBF kernel. Cross validations are employed once again to select the parameters. Classification accuracies are reported in TABLE 1 where the first row represents the source domains.

On the $adaptation$ stage, one of the four domains (B, D, K, E) is treated as target domain in turn, while the others as source domains. The one-to-one baselines are used here for adaptation.

Figure 3 shows that BayesMSDA proposed in this paper gives an encouraging result for binary classification. The BayesMSDA beats the majority voting method on three domains (B, E, K). Comparing TABLE 1 and Figure 3, we can conclude that BayesMSDA is a tradeoff between the best and the worst one-to-one linear classifiers. Because there are no labeled data at disposal in the target domain, we do not know in advance which one-to-one classifier is the best one and which is the worst. Choosing classifiers randomly is not accecptale. Our method is a better choice to get a resonable result, because even on domain D BayesMSDA outperforms the other two domains (74.1% VS 69.6% & 69.3%) except the best baseline.

4. Conclusions and future work

In this paper, a new Bayesian framework for multi-source domain adaptation (BayesMSDA) is proposed. We focus on the case that there are lots of labeled data in source domains but no labeled data are at disposal in the target domain. Our experimental results show that BayesMSDA is a better choice when no labeled data are available in the target domain.

It is also an interesting problem to consider the situation that there are some labeled instances available in the target domain. In the future, we will study this problem.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Project 61075005, and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

[1] S.B. David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J.W. Vaughan, “A Theory of Learning from Different Domains”, Machine Learning, Vol 79, pp. 151-175, 2010.

[2] W. Tu, and S. Sun, “Transferable Discriminative Dimensionality Reduction”, Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, pp. 865-868, Nov. 2011.

[3] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs Up? Sentiment Classification using Machine Learning Techniques”, Proceedings of Empirical methods in Natural Language Processing, Vol 10, pp. 79-86, 2002.

[4] J. Blitzer, M. Dredze, and F. Pereira, “Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification”, Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 440-447, Jun. 2007

[5] A. Aue, and M. Gamon, “Customizing Sentiment Classifiers to New Domains: A Case Study”, Proceedings of Recent Advances in Natural Language Processing, 2005.

[6] J. Jiang, and C. Zhai, “Instance Weighting for Domain Adaptation in NLP”, Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 264-271, Jun. 2007.

[7] K. Crammer, M. Kearns, and J. Wortman, “Learning from Multiple Source”, Journal of Machine Learning Research, Vol 9, pp. 1757-1774, Jun. 2008.

[8] W. Tu, and S. Sun, “Dynamical Ensemble Learning with Model-Friendly Classifiers for Domain Adaptation”, Proceedings of the 21st International Conference on Pattern Recognition 2012, pp. 1181-1184, Nov. 2012.

[9] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain Adaptation with Multiple Sources”, Advances in Neural Information Processing Systems, Vol 21, pp. 1041-1048, 2008.

[10] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain Adaptation: Learning Bounds and Algorithms”, Proceedings of the Conference on Learning Theory, Jun. 2009.

[11] M. Belkin, and P. Niyogi, “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation”, Neural Computation, Vol 15, pp. 1373-1396, Jun. 2003.

[12] S. Sun, “Multi-view Laplacian Support Vector Machines”, Lecture Notes in Artificial Intelligence, Vol 7121, pp. 209-222, Dec. 2011.

[13] T. Joachims, “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”, European Conference on Machine Learning, pp. 137-142, Apr. 1998.

[14] J. Shawe-Taylor, and S. Sun, “A Review of Optimization Methodologies in Support Vector Machines”, Neurocomputing, Vol 74, pp. 3609-3618, Oct. 2011.