Information Fusion 24 (2015) 84–92

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

A survey of multi-source domain adaptation

Shiliang Sun *, Honglei Shi, Yuanbin Wu

Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, 500 Dongchuan Road, Shanghai 200241, PR China

ARTICLE INFO

Article history:
Received 20 September 2014
Received in revised form 5 December 2014
Accepted 14 December 2014
Available online 20 December 2014

Keywords:
Machine learning
Multi-source learning
Domain adaptation
Transfer learning

ABSTRACT

In many machine learning algorithms, a major assumption is that the training and the test samples are in the same feature space and have the same distribution. However, for many real applications this assumption does not hold. In this paper, we survey the problem where the training samples and the test samples are from different distributions. This problem can be referred as *domain adaptation*. The training samples, always with labels, are obtained from what is called source domains, while the test samples, which usually have no labels or only a few labels, are obtained from what is called target domains. The source domains and the target domains are different but related to some extent; the learners can learn some information from the source domains for the learning of the target domains. We focus on the multi-source domain adaptation problem where there is more than one source domain available together with only one target domain. A key issue is how to select good sources and samples for the adaptation. In this survey, we review some theoretical results and well developed algorithms for the multi-source domain adaptation problem. We also discuss some open problems which can be explored in future work.

1. Introduction

In machine learning, most models such as Gaussian process (GP), linear discriminative analysis (LDA), support vector machine (SVM) [1,2] and principal component analysis (PCA), assume that training samples are drawn according to the same distribution as the unseen test samples. Uniform convergence theory guarantees that a model's empirical training error is close to its true error with high probability. However, there are many cases in practice where the training and the test distributions differ. We wish to train a model in one or more domains (called source domains) and then apply it to another different but related domain (called target domain). Such learning task is known as *domain adaptation* [3–8], which is confronted in many applications, like computer vision [9–12], sentimental analysis [13–16], natural language processing [17], video concept detection [18,19], and wifi localization detection [20]. In these problems, users are generally reluctant to annotate abundant samples (like consumer videos, or the reviews for certain products) to train an effective model for later classification. What they have are a set of limited labeled samples and a large number of unlabeled data. The task is to combine the labeled source data and unlabeled target data to classify the target data as correctly as possible. The difficulty lies in the mismatch between the source distribution and the target distribution. Domain adaptation approaches explicitly or implicitly handle the mismatch between data distributions of the source and target domains.

Domain adaptation is one of the branches of transfer learning. According to Pan et al. [8], transductive transfer learning can be categorized into two cases. The first case is that the feature spaces between the source and target domains are different, i.e. $\mathcal{X}_S \neq \mathcal{X}_T$. The second case is that the feature spaces between the source and target domains are the same, but the marginal probability distributions of the input data are different, i.e. $\mathcal{X}_S = \mathcal{X}_T$, but $P(\mathcal{X}_S) \neq P(\mathcal{X}_T)$. The latter case can be referred as *domain adaptation*. Domain adaptation is different from semi-supervised learning and data set shift [21]. It assumes that the labeled and unlabeled data come from different but related domains, while semi-supervised learning methods employ both labeled and unlabeled data from the same domain. On the other hand, data set shift assumes that the joint distribution $P(\mathcal{X}, \mathcal{Y})$ of input $\mathcal{X}$ and output $\mathcal{Y}$ changes across the source and target domains, i.e. $P(\mathcal{X}, \mathcal{Y})_S \neq P(\mathcal{X}, \mathcal{Y})_T$. However, the focus of domain adaptation is that the marginal probability distributions of the input data are different.

For the single-source domain setting, much work has been developed. Several theoretical analyses have considered the single-source domain adaptation problem. Ben-David et al. [22] defined two sources of adaptation errors. Firstly, feature distributions differ between the source and the target domains, which means that the test examples are different from the training

* Corresponding author. Tel.: +86 21 54345183; fax: +86 21 54345119.
E-mail addresses: shiliangsun@gmail.com, slsun@cs.ecnu.edu.cn (S. Sun).

http://dx.doi.org/10.1016/j.inffus.2014.12.003
1566-2535/

examples in the sense of data distributions. Since many applications usually use the lexical items as features, this problem can be especially difficult. In general, this problem can be addressed by using the unlabeled target data since feature distributions can be measured and aligned without annotated examples. Secondly, the decision functions differ between domains. The instance may be labeled differently depending on the domains. To correct this error, one needs the knowledge of the labeling function, which can only be gained from labeled target samples. Dredze et al. [23] showed how domain adaptation for parsing is difficult when annotation guidelines differ for different domains.

In addition to the theoretical analyses, there is also much empirical work on algorithms for single-source domain adaptation. Chelba and Acero [24] trained a classifier on the source domain, and then used the maximum a posteriori (MAP) estimation of the weights of a maximum entropy target domain classifier. The prior is a Gaussian distribution whose mean is equal to the weights of the source domain classifier. Daume and Marcu [25] used an empirical Bayes model to estimate a latent variable model which groups the instances into two categories domain-specific or common across both domains. Blitzer et al. [26] introduced structural correspondence learning to automatically induce correspondences among features from two domains, without using the labeled target data. Unlike the work of Daume and Marcu [25], they found a common representation for features from different domains, rather than instances.

Often in practice, one may be offered more than one source domain for training. It is wasteful if we only use one source for training. The most common way is to add up all the sources as one source. However, this approach ignores the difference among the sources. A second way is to train a classifier per source and combine these multiple base classifiers. Based on the principles of risk minimization, one can derive a solution which assigns weights for each base model, and combines multiple base models to maximize their combined accuracy on the new domain. The combined model can get a reasonable high accuracy for the target task. The second model for multi-source domain adaptation is displayed in Fig. 1.

One popular domain adaptation problem arises in text classification tasks where one can retrieve information from several source domains and make predictions about another target domain. In natural language processing, sentiment classification is a task of classifying documents according to the sentiments. Given a piece of text (usually a review or essay), what is of interest is whether the opinion expressed by the text is positive or negative. Sentiment analysis is useful on a number of text domains, ranging from stock message boards to congressional floor debates. In some domains (e.g. movie reviews and book reviews), one can

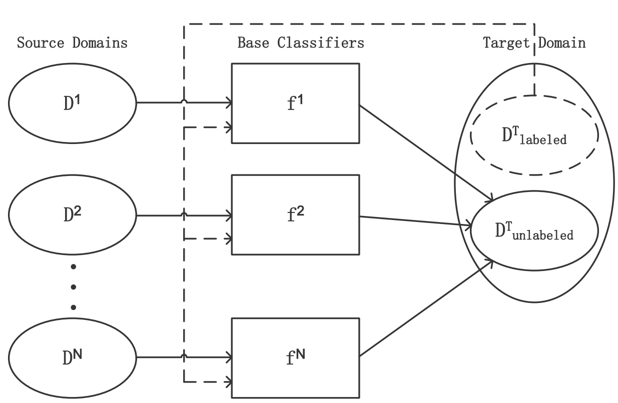

Fig. 1. The model for multi-source domain adaptation.

have plenty of labeled data for machine learning algorithms to train a model for classification, while there are also many domains (e.g. piano reviews) that can not have enough data available for training. Domain adaptation algorithms can solve such problems by using the domains which have plenty of labeled as sources, and domains lack of labeled data as target domains. Usually, the source and target domains are assumed to be different but related.

Another domain adaptation application is the computational advertising system. The system may rank advertisements for queries originating from many different countries, in many different languages, and covering a variety of product domains. A system trained on all queries together, agnostic with respect to such properties, may benefit from having a large quantity of training data. However, it is also possible that data sources have conflicting properties that reduce the performance of a single model trained in this manner. In this case, it would be preferable to train separate systems. In fact, both approaches are inadequate. Data sources typically share some common characteristics and behaviors, though differ from one another. A single system obscures differences, while separate systems ignore similarities.

Besides the above applications, some efforts have also been made on domain adaptation for event recognition in consumer videos [18,10]. For example, Duan et al. [10] learned a classifier which uses both the SIFT features of web images from source domains and the space–time (ST) features as well as SIFT features from the target domain to make decisions for the target video.

In this paper, we investigate both theoretical analyses and existing algorithms for multi-source domain adaptation. We hope to provide a useful resource for the research of multi-source domain adaptation. The rest of the survey is organized as follows. In Section 2, some theoretical analyses are provided. Section 3 covers some well-developed algorithms. We summarize Sections 2 and 3 in Table 1 for a quick access to the methods introduced in this paper. In Section 4, some performance evaluation measurements as well as publicly available datasets about multi-source domain adaptation are listed. Conclusions and some worth-working lines for multi-source domain adaptation are summarized in Section 5.

2. Theoretical analyses for multi-source domain adaptation

We formalize the multi-source domain adaptation problem as follows. Let $\mathcal{X}$ be the input space, D be a distribution on $\mathcal{X}$, and $f : \mathcal{X} \to \mathbb{R}$ be the target function to learn. A domain is defined as a pair $\langle D, f\rangle$. Let $\mathcal{L}(f(x), y) \in \mathbb{R}$ be a loss function with respect to f. Suppose we have N distinct sources, with each source S_j associated with an unknown distribution D_j over the input points, and an unknown labeling function f_j. Each source S_j has $m_j = \eta_j m$ labeled samples where m is the total sample number from all the sources, and $\eta_j \in [0, 1]$, $\sum \eta_j = 1$. The objective is to use these samples to train a model to perform well on a target domain $\langle D_T, f_T\rangle$. The multi-source domain adaptation problem is to combine each source S_j to derive a hypothesis h with a small loss $\mathcal{L}(f_T(x), h(x))$ on the target domain.

Blitzer et al. [40] gave a bound on the error rate of a hypothesis derived from a weighted combination of the source data sets for the specific case of empirical risk minimization.

Crammer et al. [27] addressed a problem where multiple sources are present. But the nature of the problem differs from adaptation since the distribution of the input points is the same for all these sources, and only the labels change due to the varying amounts of noise. They gave a general bound on the expected loss of the model by minimizing the empirical loss on the nearest k sources. These nearest k sources form a recommended set of sources. Two key ingredients needed to apply this bound were