A Review About Transfer Learning Methods and Applications

Yongqiang Liu
Library Huaihai Institute of Technology, Lianyungang, China

Abstract. The machine learning plays the key roles in many artificial intelligence areas including classification, regression and clustering. The traditional machine learning methods assume that the training and test sample are drawn from the same feature space and the same distribution. With the change of the distribution, the traditional machine learning methods need to rebuild the models using newly collected training samples. In real world, it is impossible or expensive to recollect and label the needed training samples and rebuild the models. To address the problem, the transfer learning is proposed. This paper reviews four categories of transfer learning methods and their applications: transferring instances, transferring feature representations, transferring parameters and transferring relationship.

Keywords: review, transfer learning, machine learning

1. Introduction

The machine learning plays the key roles in many artificial intelligence areas including classification, regression and clustering. The traditional machine learning methods assume that the training and test sample are drawn from the same feature space and the same distribution. With the change of the distribution, the traditional machine learning methods need to rebuild the models using newly collected training samples. In real world, it is impossible or expensive to recollect and label the needed training samples and rebuild the models. To address the problem, the transfer learning is proposed.

One example is Web-document classification [1], [2], where our goal is to classify a given Web document into several predefined categories. As an example, in the area of Web document classification (see, e.g., [3]), the labeled examples may be the university webpages that are associated with category information obtained through previous manual labelling efforts.

In this survey paper, we give a comprehensive overview of transfer learning for classification, regression, and clustering developed in machine learning and data mining areas. There has been a large amount of work on transfer learning for reinforcement learning in the machine learning literature (e.g., [4]). However, in this paper, we only focus on transfer learning for classification, regression, and clustering problems that are related more closely to data mining tasks. By doing the survey, we hope to provide a useful resource for the data mining and machine learning community. The detailed review can be seen in [5].

2. The Review on Transfer Learning

2.1. Transfer Instances

The instance-transfer approach to the inductive transfer learning setting is intuitively appealing: although the source domain data cannot be reused directly, there are certain parts of the data that can still be reused together with a few labeled data in the target domain.

Dai et al. [3] proposed a boosting algorithm, TrAdaBoost, which is an extension of the AdaBoost algorithm, to address the inductive transfer learning problems. TrAdaBoost assumes that the source and target-domain data use exactly the same set of features and labels, but the distributions of the data in the two domains are different. In addition, TrAdaBoost assumes that, due to the difference in distributions between
the source and some of the source domain data may be useful in learning for the target domain but some of them may not and could even be harmful. It attempts to iteratively reweight the source domain data to reduce the effect of the “bad” source data while encourage the “good” source data to contribute more for the target domain. For each round of iteration, TrAdaBoost trains the base classifier on the weighted source and target data. The error is only calculated on the target data. Furthermore, TrAdaBoost uses the same strategy as AdaBoost to update the incorrectly classified examples in the target domain while using a different strategy from AdaBoost to update the incorrectly classified source examples in the source domain.

More formally, let \( X_s \) be the same-distribution instance space, \( X_d \) be the diff-distribution instance space, and \( Y = 0, 1 \) be the set of category labels. A concept is a boolean function \( c \) mapping from \( X \) to \( Y \), where \( X = X_s \cup X_d \). The test data set is denoted by \( S = \{(x'_i)\} \), where \( x'_i \in X_j (i = 1, \ldots, k) \). Here, \( k \) is the size of the test set \( S \) which is unlabeled. The training data set \( T \subseteq \{X \times Y\} \) is partitioned into two labeled sets \( T_d \) and \( T_s \). \( T_d \) represents the diff-distribution training data that \( T_d = \{(x'_i, c(x'_i))\} \), where \( x'_i \in X_s (i = 1, \ldots, n) \). \( T_s \) represents the same-distribution training data that \( T_s = \{(x'_j, c(x'_j))\} \), where \( x'_j \in X_d (j = 1, \ldots, m) \). \( n \) and \( m \) are the sizes of \( T_d \) and \( T_s \), respectively. \( c(x) \) returns the label for the data instance \( x \). The combined training set \( T = \{(x'_i, c(x'_i))\} \) is defined as follows

\[
x_i = \begin{cases} x'_d & i = 1, \ldots, n; \\ x'_s & i = n+1, \ldots, n+m. \end{cases}
\]

Here, \( T_d \) corresponds to some labeled data from an old domain that we try to reuse as much as we can; however we do not know which part of \( T_d \) is useful to us.

In each iteration round, if a diff-distribution training instance is mistakenly predicted, the instance may likely conflict with the same-distribution training data. Then, we decrease its training weight to reduce its effect through multiplying its weight by \( \beta^{\hat{h}(x_i)-c(x_i)} \). Note that \( \beta^{\hat{h}(x_i)-c(x_i)} \in (0, 1) \). Thus, in the next round, the misclassified diff-distribution training instances, which are dissimilar to the same-distribution ones, will affect the learning process less than the current round. After several iterations, the diff-distribution training instances that fit the same-distribution ones better will have larger training weights, while the diff-distribution training instances that are dissimilar to the same-distribution ones will have lower weights. The instances with large training weights will intend to help the learning algorithm to train better classifiers.

### 2.2. Transfer Feature Representations

The feature-representation-transfer approach to the inductive transfer learning problem aims at finding “good” feature representations to minimize domain divergence and classification or regression model error. Strategies to find “good” feature representations are different for different types of the source domain data. If a lot of labeled data in the source domain are available, supervised learning methods can be used to construct a feature representation. This is similar to common feature learning in the field of multitask learning [6]. If no labeled data in the source domain are available, unsupervised learning methods are proposed to construct the feature representation.

Supervised feature construction methods for the inductive transfer learning setting are similar to those used in multitask learning. The basic idea is to learn a low-dimensional representation that is shared across related tasks. In addition, the learned new representation can reduce the classification or regression model error of each task as well. Argyriou et al. [6] proposed a sparse feature learning method for multitask learning. In the inductive transfer learning setting, the common features can be learned by solving an optimization problem, given as follows:

\[
\min_{A, U} \sum_{i \in \{T, S\}} \sum_{j=1}^{n_i} L(y_i^j, <a_i, U_j x_i^j>) + \gamma \| A \|^{2}_{r, p} \tag{1}
\]

In this equation, \( S \) and \( T \) denote the tasks in the source domain and target domain, respectively. \( A = [a_s, a_t] \in \mathbb{R}^{d \times 2} \) is a matrix of parameters. \( U \) is a \( d \times d \) orthogonal matrix (mapping function) for mapping the original high-dimensional data to low-dimensional representations. The \( (r, p) \)-norm of \( A \) is defined as \( \| A \|_{r, p} = \left( \sum_{i=1}^{d} |a_{ij}|^p \right)^{\frac{1}{p}} \). The optimization problem \((\text{ref\{eq:mt\}})\) estimates the low-dimensional...
representations $U^T X_T$, $U^T X_S$ and the parameters, $A$, of the model at the same time. The optimization problem (1) can be further transformed into an equivalent convex optimization formulation and be solved efficiently. In a follow-up work, Argyriou et al. [7] proposed a spectral regularization framework on matrices for multitask structure learning.

### 2.3. Transfer Parameters

Most parameter-transfer approaches to the inductive transfer learning setting assume that individual models for related tasks should share some parameters or prior distributions of hyperparameters. Most approaches described in this section, including a regularization framework and a hierarchical Bayesian framework, are designed to work under multitask learning. However, they can be easily modified for transfer learning. As mentioned above, multitask learning tries to learn both the source and target tasks simultaneously and perfectly, while transfer learning only aims at boosting the performance of the target domain by utilizing the source domain data. Thus, in multitask learning, weights of the loss functions for the source and target data are the same. In contrast, in transfer learning, weights in the loss functions for different domains can be different. Intuitively, we may assign a larger weight to the loss function of the target domain to make sure that we can achieve better performance in the target domain.

Lawrence and Platt [8] proposed an efficient algorithm known as MT-IVM, which is based on Gaussian Processes (GP), to handle the multitask learning case. MT-IVM tries to learn parameters of a Gaussian Process over multiple tasks by sharing the same GP prior. Bonilla et al. [9] also investigated multitask learning in the context of GP. The authors proposed to use a free-form covariance matrix over tasks to model intertask dependencies, where a GP prior is used to induce correlations between tasks. Schwaighofer et al. [10] proposed to use a hierarchical Bayesian framework (HB) together with GP for multitask learning.

Besides transferring the priors of the GP models, some researchers also proposed to transfer parameters of SVMs under a regularization framework. Evgeniou and Pontil [11] borrowed the idea of HB to SVMs for multitask learning. The proposed method assumed that the parameter, $w$, in SVMs for each task can be separated into two terms. One is a common term over tasks and the other is a task-specific term. In inductive transfer learning,

$$w_S = w_0 + v_S, \quad w_T = w_0 + v_T,$$

where $w_S$ and $w_T$ are parameters of the SVMs for the source task and the target learning task, respectively. $w_0$ is a common parameter while $v_S$ and $v_T$ are specific parameters for the source task and the target task, respectively. By assuming $f_t = w_t x$ to be a hyperplane for task $t$, an extension of SVMs to multitask learning case can be written as the following:

$$\min_{w_0, v_S, v_T} J(w_0, v_S, v_T) = \sum_{n \in \{S,T\}} \sum_{i=1}^{n} \xi_i + \frac{\lambda}{2} \sum_{n \in \{S,T\}} \|v_n\|^2 + \lambda_2 \| w_0 \|^2$$

By solving the optimization problem above, we can learn the parameters $w_0$, $v_S$, and $v_T$ simultaneously.

The parameter-transfer approach was applied on age estimation. Automatic age estimation from facial images has aroused research interests in recent years due to its promising potential for some computer vision applications. The age estimation methods can be grouped into two categories: global age estimation and personalized age estimation. Global age estimation is based on the assumption that the aging process is the same for all people and hence the same age estimator can be used for different people. On the other hand, personalized age estimation is based on the assumption that different people go through different aging processes and hence different (personalized) age estimators are needed. Their experimental results show that personalized age estimation generally outperforms global age estimation.

Multi-task learning [12], [13] is a learning paradigm which seeks to improve the generalization performance of a learning task with the help of some other related tasks. This learning paradigm has been inspired by human learning activities in that people often apply the knowledge gained from previous learning tasks to help learn a new task. For example, a baby first learns to recognize human faces and later uses this knowledge to help it learn to recognize other objects. Multi-task learning can help to alleviate the small sample size problem which arises when each learning task contains only a very limited number of training data points. One widely adopted approach in multi-task learning is to learn a common data or model...
representation from multiple tasks [8] by leveraging the relationships between tasks. In this paper, we propose a novel approach to age estimation based on this idea from multi-task learning.

In age estimation, even though there exist (possibly substantial) differences between the aging processes of different individuals, some common patterns are still shared by them. For example, the face as a whole and all the facial parts will become bigger as one grows from a baby to an adult. Also, facial wrinkles generally increase as one gets older. As such, one may distinguish facial features of the aging process into two types. The first type is common to essentially all persons and the second type is person specific. From the perspective of multi-task learning, we want to use data from all tasks to learn the common features while task-specific (person-specific) features are learned separately for each task (person). Thus, age estimation can be formulated as a multi-task regression problem in which each learning task refers to estimation of the age function of each person. Suppose the age function of the \( i \) th person is approximated by the age estimator \( h_i(x; \alpha, \beta_i) \), where \( \alpha \) and \( \beta_i \) are the common and person-specific model parameters corresponding to the two types of facial features. The learning problem thus corresponds to learning parameters of both types.

The only age estimation method related to ours is WAS [14] in which the age estimator for the \( i \) th person has the form \( g_i(x; \gamma_i) \), where \( \gamma_i \) models all the features of the \( i \) th person including features common to all persons and person-specific ones. Since \( \gamma_i \) includes all features, it is generally of high dimensionality. However, \( \gamma_i \) is estimated using only data for the \( i \) th person which is typically very limited (e.g., below 20 for the FG-NET database and about 3 for the MORPH database). As a result, it is difficult to estimate \( \gamma_i \) accurately. In our approach, since the common features are represented in \( \alpha \), \( \beta_i \) which captures the person-specific features for the \( i \) th person only has low dimensionality. Thus \( \beta_i \) can be estimated more accurately even though each task has very limited training data. Fig. 1 below summarizes the differences between WAS and our method. Moreover, since we model age estimation as a regression problem, we do not need to estimate the missing parts in the aging pattern and hence can overcome the limitations of AGES [15].

![Diagram](image.png)

**Fig. 1:** Modeling processes of WAS (left) and our method (right). In WAS, the age estimator \( g_i \) for the \( i \) th person is parameterized by an independent parameter vector \( \gamma_i \). In our method, the age estimator \( h_i \) for the \( i \) th person is parameterized by a common parameter vector \( \alpha \) and a task-specific parameter vector \( \beta_i \).

It is worth noting that the multi-task formulation may also be used for other variants of the age estimation problem. For example, if there is no identity information in the age database so that we do not know which image belongs to which person, we may generalize the concept of a task to other available information, such as gender or ethnicity. Taking gender information for example, we may assume that all male individuals share a common aging process and all female individuals share another common aging process. Thus one task corresponds to age estimation for male while another task for female. In this paper, we only consider the setting in which each learning task is for one person. Other variants will be investigated in our future research.

### 2.4. Transfer Relational Knowledge

Different from other three contexts, the relational-knowledge-transfer approach deals with transfer learning problems in relational domains, where the data are non-i.i.d. and can be represented by multiple
relations, such as networked data and social network data. This approach does not assume that the data drawn from each domain be independent and identically distributed (i.i.d.) as traditionally assumed. It tries to transfer the relationship among data from a source domain to a target domain. In this context, statistical relational learning techniques are proposed to solve these problems.

3. Conclusions

In this survey paper, we have reviewed several current trends of transfer learning. Transfer learning is classified to three different settings: inductive transfer learning, transductive transfer learning, and unsupervised transfer learning. Most previous works focused on the former two settings. Unsupervised transfer learning may attract more and more attention in the future.

4. References