

# A Survey on Transfer Learning

Sinno Jialin Pan and Qiang Yang, *Fellow, IEEE*

**Abstract**—A major assumption in many machine learning and data mining algorithms is that the training and future data must be in the same feature space and have the same distribution. However, in many real-world applications, this assumption may not hold. For example, we sometimes have a classification task in one domain of interest, but we only have sufficient training data in another domain of interest, where the latter data may be in a different feature space or follow a different data distribution. In such cases, knowledge transfer, if done successfully, would greatly improve the performance of learning by avoiding much expensive data-labeling efforts. In recent years, transfer learning has emerged as a new learning framework to address this problem. This survey focuses on categorizing and reviewing the current progress on transfer learning for classification, regression, and clustering problems. In this survey, we discuss the relationship between transfer learning and other related machine learning techniques such as domain adaptation, multitask learning and sample selection bias, as well as covariate shift. We also explore some potential future issues in transfer learning research.

**Index Terms**—Transfer learning, survey, machine learning, data mining.

## 1 INTRODUCTION

DATA mining and machine learning technologies have already achieved significant success in many knowledge engineering areas including classification, regression, and clustering (e.g., [1], [2]). However, many machine learning methods work well only under a common assumption: the training and test data are drawn from the same feature space and the same distribution. When the distribution changes, most statistical models need to be rebuilt from scratch using newly collected training data. In many real-world applications, it is expensive or impossible to recollect the needed training data and rebuild the models. It would be nice to reduce the need and effort to recollect the training data. In such cases, *knowledge transfer* or *transfer learning* between task domains would be desirable.

Many examples in knowledge engineering can be found where transfer learning can truly be beneficial. One example is Web-document classification [3], [4], [5], 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., [6]), the labeled examples may be the university webpages that are associated with category information obtained through previous manual-labeling efforts. For a classification task on a newly created website where the data features or data distributions may be different, there may be a lack of labeled training data. As a result, we may not be able to directly apply the webpage classifiers learned on the university website to the new website. In such cases, it would be helpful if we could transfer the classification knowledge into the new domain.

The need for transfer learning may arise when the data can be easily outdated. In this case, the labeled data obtained in one time period may not follow the same distribution in a later time period. For example, in indoor WiFi localization problems, which aims to detect a user's current location based on previously collected WiFi data, it is very expensive to calibrate WiFi data for building localization models in a large-scale environment, because a user needs to label a large collection of WiFi signal data at each location. However, the WiFi signal-strength values may be a function of time, device, or other dynamic factors. A model trained in one time period or on one device may cause the performance for location estimation in another time period or on another device to be reduced. To reduce the recalibration effort, we might wish to adapt the localization model trained in one time period (the source domain) for a new time period (the target domain), or to adapt the localization model trained on a mobile device (the source domain) for a new mobile device (the target domain), as done in [7].

As a third example, consider the problem of sentiment classification, where our task is to automatically classify the reviews on a product, such as a brand of camera, into positive and negative views. For this classification task, we need to first collect many reviews of the product and annotate them. We would then train a classifier on the reviews with their corresponding labels. Since the distribution of review data among different types of products can be very different, to maintain good classification performance, we need to collect a large amount of labeled data in order to train the review-classification models for each product. However, this data-labeling process can be very expensive to do. To reduce the effort for annotating reviews for various products, we may want to adapt a classification model that is trained on some products to help learn classification models for some other products. In such cases, transfer learning can save a significant amount of labeling effort [8].

In this survey paper, we give a comprehensive overview of transfer learning for classification, regression, and cluster-

• The authors are with the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong.  
E-mail: {sinnopan, qyang}@cse.ust.hk.

Manuscript received 13 Nov. 2008; revised 29 May 2009; accepted 13 July 2009; published online 12 Oct. 2009.

Recommended for acceptance by C. Clifton.

For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2008-11-0600. Digital Object Identifier no. 10.1109/TKDE.2009.191.

ing 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., [9], [10]). 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 rest of the survey is organized as follows: In the next four sections, we first give a general overview and define some notations we will use later. We, then, briefly survey the history of transfer learning, give a unified definition of transfer learning and categorize transfer learning into three different settings (given in Table 2 and Fig. 2). For each setting, we review different approaches, given in Table 3 in detail. After that, in Section 6, we review some current research on the topic of “negative transfer,” which happens when knowledge transfer has a negative impact on target learning. In Section 7, we introduce some successful applications of transfer learning and list some published data sets and software toolkits for transfer learning research. Finally, we conclude the paper with a discussion of future works in Section 8.

## 2 OVERVIEW

### 2.1 A Brief History of Transfer Learning

Traditional data mining and machine learning algorithms make predictions on the future data using statistical models that are trained on previously collected labeled or unlabeled training data [11], [12], [13]. Semisupervised classification [14], [15], [16], [17] addresses the problem that the labeled data may be too few to build a good classifier, by making use of a large amount of unlabeled data and a small amount of labeled data. Variations of supervised and semisupervised learning for imperfect data sets have been studied; for example, Zhu and Wu [18] have studied how to deal with the noisy class-label problems. Yang et al. considered cost-sensitive learning [19] when additional tests can be made to future samples. Nevertheless, most of them assume that the distributions of the labeled and unlabeled data are the same. *Transfer learning*, in contrast, allows the domains, tasks, and distributions used in training and testing to be different. In the real world, we observe many examples of transfer learning. For example, we may find that learning to recognize apples might help to recognize pears. Similarly, learning to play the electronic organ may help facilitate learning the piano. The study of *Transfer learning* is motivated by the fact that people can intelligently apply knowledge learned previously to solve new problems faster or with better solutions. The fundamental motivation for *Transfer learning* in the field of machine learning was discussed in a NIPS-95 workshop on “Learning to Learn,”<sup>1</sup> which focused on the need for lifelong machine learning methods that retain and reuse previously learned knowledge.

Research on transfer learning has attracted more and more attention since 1995 in different names: learning to learn, life-long learning, knowledge transfer, inductive

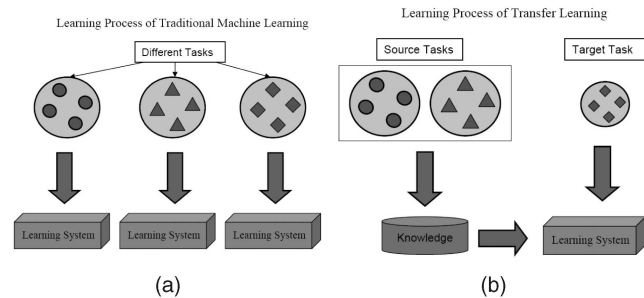


Fig. 1. Different learning processes between (a) traditional machine learning and (b) transfer learning.

transfer, multitask learning, knowledge consolidation, context-sensitive learning, knowledge-based inductive bias, metalearning, and incremental/cumulative learning [20]. Among these, a closely related learning technique to transfer learning is the multitask learning framework [21], which tries to learn multiple tasks simultaneously even when they are different. A typical approach for multitask learning is to uncover the common (latent) features that can benefit each individual task.

In 2005, the Broad Agency Announcement (BAA) 05-29 of Defense Advanced Research Projects Agency (DARPA)’s Information Processing Technology Office (IPTO)<sup>2</sup> gave a new mission of transfer learning: the ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks. In this definition, *transfer learning* aims to extract the knowledge from one or more *source tasks* and applies the knowledge to a *target task*. In contrast to multitask learning, rather than learning all of the source and target tasks simultaneously, transfer learning cares most about the target task. The roles of the source and target tasks are no longer symmetric in transfer learning.

Fig. 1 shows the difference between the learning processes of traditional and transfer learning techniques. As we can see, traditional machine learning techniques try to learn each task from scratch, while transfer learning techniques try to transfer the knowledge from some previous tasks to a target task when the latter has fewer high-quality training data.

Today, transfer learning methods appear in several top venues, most notably in data mining (ACM KDD, IEEE ICDM, and PKDD, for example), machine learning (ICML, NIPS, ECML, AAAI, and IJCAI, for example) and applications of machine learning and data mining (ACM SIGIR, WWW, and ACL, for example).<sup>3</sup> Before we give different categorizations of transfer learning, we first describe the notations used in this paper.

### 2.2 Notations and Definitions

In this section, we introduce some notations and definitions that are used in this survey. First of all, we give the definitions of a “domain” and a “task,” respectively.

In this survey, a *domain*  $\mathcal{D}$  consists of two components: a feature space  $\mathcal{X}$  and a marginal probability distribution  $P(X)$ , where  $X = \{x_1, \dots, x_n\} \in \mathcal{X}$ . For example, if our learning task

2. <http://www.darpa.mil/ipto/programs/tl/tl.asp>.

3. We summarize a list of conferences and workshops where transfer learning papers appear in these few years in the following webpage for reference, <http://www.cse.ust.hk/~sinnopan/conferenceTL.htm>.

1. [http://socrates.acadiau.ca/courses/comp/dsilver/NIPS95\\_LTL/transfer.workshop.1995.html](http://socrates.acadiau.ca/courses/comp/dsilver/NIPS95_LTL/transfer.workshop.1995.html).

TABLE 1  
Relationship between Traditional Machine Learning and Various Transfer Learning Settings

Learning Settings		Source and Target Domains	Source and Target Tasks
Traditional Machine Learning		the same	the same
Transfer Learning	<i>Inductive Transfer Learning /</i>	the same	different but related
	<i>Unsupervised Transfer Learning</i>	different but related	different but related
	<i>Transductive Transfer Learning</i>	different but related	the same

is document classification, and each term is taken as a binary feature, then  $\mathcal{X}$  is the space of all term vectors,  $x_i$  is the  $i$ th term vector corresponding to some documents, and  $X$  is a particular learning sample. In general, if two domains are different, then they may have different feature spaces or different marginal probability distributions.

Given a specific domain,  $\mathcal{D} = \{\mathcal{X}, P(X)\}$ , a *task* consists of two components: a label space  $\mathcal{Y}$  and an objective predictive function  $f(\cdot)$  (denoted by  $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ ), which is not observed but can be learned from the training data, which consist of pairs  $\{x_i, y_i\}$ , where  $x_i \in X$  and  $y_i \in \mathcal{Y}$ . The function  $f(\cdot)$  can be used to predict the corresponding label,  $f(x)$ , of a new instance  $x$ . From a probabilistic viewpoint,  $f(x)$  can be written as  $P(y|x)$ . In our document classification example,  $\mathcal{Y}$  is the set of all labels, which is True, False for a binary classification task, and  $y_i$  is “True” or “False.”

For simplicity, in this survey, we only consider the case where there is one source domain  $\mathcal{D}_S$ , and one target domain,  $\mathcal{D}_T$ , as this is by far the most popular of the research works in the literature. More specifically, we denote the *source domain data* as  $D_S = \{(x_{S_1}, y_{S_1}), \dots, (x_{S_{n_S}}, y_{S_{n_S}})\}$ , where  $x_{S_i} \in \mathcal{X}_S$  is the data instance and  $y_{S_i} \in \mathcal{Y}_S$  is the corresponding class label. In our document classification example,  $D_S$  can be a set of term vectors together with their associated true or false class labels. Similarly, we denote the *target-domain data* as  $D_T = \{(x_{T_1}, y_{T_1}), \dots, (x_{T_{n_T}}, y_{T_{n_T}})\}$ , where the input  $x_{T_i}$  is in  $\mathcal{X}_T$  and  $y_{T_i} \in \mathcal{Y}_T$  is the corresponding output. In most cases,  $0 \leq n_T \ll n_S$ .

We now give a unified definition of transfer learning.

**Definition 1 (Transfer Learning).** *Given a source domain  $\mathcal{D}_S$  and learning task  $\mathcal{T}_S$ , a target domain  $\mathcal{D}_T$  and learning task  $\mathcal{T}_T$ , transfer learning aims to help improve the learning of the target predictive function  $f_T(\cdot)$  in  $\mathcal{D}_T$  using the knowledge in  $\mathcal{D}_S$  and  $\mathcal{T}_S$ , where  $\mathcal{D}_S \neq \mathcal{D}_T$ , or  $\mathcal{T}_S \neq \mathcal{T}_T$ .*

In the above definition, a domain is a pair  $\mathcal{D} = \{\mathcal{X}, P(X)\}$ . Thus, the condition  $\mathcal{D}_S \neq \mathcal{D}_T$  implies that either  $\mathcal{X}_S \neq \mathcal{X}_T$  or  $P_S(X) \neq P_T(X)$ . For example, in our document classification example, this means that between a source document set and a target document set, either the term features are different between the two sets (e.g., they use different languages), or their marginal distributions are different.

Similarly, a task is defined as a pair  $\mathcal{T} = \{\mathcal{Y}, P(Y|X)\}$ . Thus, the condition  $\mathcal{T}_S \neq \mathcal{T}_T$  implies that either  $\mathcal{Y}_S \neq \mathcal{Y}_T$  or  $P(Y_S|X_S) \neq P(Y_T|X_T)$ . When the target and source domains are the same, i.e.,  $\mathcal{D}_S = \mathcal{D}_T$ , and their learning tasks are the same, i.e.,  $\mathcal{T}_S = \mathcal{T}_T$ , the learning problem becomes a traditional machine learning problem. When the domains are different, then either 1) the feature spaces between the domains are different, i.e.,  $\mathcal{X}_S \neq \mathcal{X}_T$ , or 2) the feature spaces between the domains are the same but the marginal

probability distributions between domain data are different; i.e.,  $P(X_S) \neq P(X_T)$ , where  $X_{S_i} \in \mathcal{X}_S$  and  $X_{T_i} \in \mathcal{X}_T$ . As an example, in our document classification example, case 1 corresponds to when the two sets of documents are described in different languages, and case 2 may correspond to when the source domain documents and the target-domain documents focus on different topics.

Given specific domains  $\mathcal{D}_S$  and  $\mathcal{D}_T$ , when the learning tasks  $\mathcal{T}_S$  and  $\mathcal{T}_T$  are different, then either 1) the label spaces between the domains are different, i.e.,  $\mathcal{Y}_S \neq \mathcal{Y}_T$ , or 2) the conditional probability distributions between the domains are different; i.e.,  $P(Y_S|X_S) \neq P(Y_T|X_T)$ , where  $Y_{S_i} \in \mathcal{Y}_S$  and  $Y_{T_i} \in \mathcal{Y}_T$ . In our document classification example, case 1 corresponds to the situation where source domain has binary document classes, whereas the target domain has 10 classes to classify the documents to. Case 2 corresponds to the situation where the source and target documents are very unbalanced in terms of the user-defined classes.

In addition, when there exists some relationship, explicit or implicit, between the feature spaces of the two domains, we say that the source and target domains are *related*.

### 2.3 A Categorization of Transfer Learning Techniques

In *transfer learning*, we have the following three main research issues: 1) what to transfer, 2) how to transfer, and 3) when to transfer.

“What to transfer” asks which part of knowledge can be transferred across domains or tasks. Some knowledge is specific for individual domains or tasks, and some knowledge may be common between different domains such that they may help improve performance for the target domain or task. After discovering which knowledge can be transferred, learning algorithms need to be developed to transfer the knowledge, which corresponds to the “how to transfer” issue.

“When to transfer” asks in which situations, transferring skills should be done. Likewise, we are interested in knowing in which situations, knowledge should **not** be transferred. In some situations, when the source domain and target domain are not related to each other, brute-force transfer may be unsuccessful. In the worst case, it may even hurt the performance of learning in the target domain, a situation which is often referred to as *negative transfer*. Most current work on transfer learning focuses on “What to transfer” and “How to transfer,” by implicitly assuming that the source and target domains be related to each other. However, how to avoid negative transfer is an important open issue that is attracting more and more attention in the future.

Based on the definition of transfer learning, we summarize the relationship between traditional machine learning and various transfer learning settings in Table 1, where we

TABLE 2  
Different Settings of Transfer Learning

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
<i>Inductive Transfer Learning</i>	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
<i>Transductive Transfer Learning</i>	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
<i>Unsupervised Transfer Learning</i>		Unavailable	Unavailable	Clustering, Dimensionality Reduction

categorize transfer learning under three subsettings, *inductive transfer learning*, *transductive transfer learning*, and *unsupervised transfer learning*, based on different situations between the source and target domains and tasks.

1. In the *inductive transfer learning* setting, the target task is different from the source task, no matter when the source and target domains are the same or not.

In this case, some labeled data in the target domain are required to *induce* an objective predictive model  $f_T(\cdot)$  for use in the target domain. In addition, according to different situations of labeled and unlabeled data in the source domain, we can further categorize the *inductive transfer learning* setting into two cases:

- a. A lot of labeled data in the source domain are available. In this case, the *inductive transfer learning* setting is similar to the multitask learning setting. However, the *inductive transfer learning* setting only aims at achieving high performance in the target task by transferring knowledge from the source task while multitask learning tries to learn the target and source task simultaneously.
  - b. No labeled data in the source domain are available. In this case, the *inductive transfer learning* setting is similar to the *self-taught learning* setting, which is first proposed by Raina et al. [22]. In the self-taught learning setting, the label spaces between the source and target domains may be different, which implies the side information of the source domain cannot be used directly. Thus, it's similar to the inductive transfer learning setting where the labeled data in the source domain are unavailable.
2. In the *transductive transfer learning* setting, the source and target tasks are the same, while the source and target domains are different.

In this situation, no labeled data in the target domain are available while a lot of labeled data in the source domain are available. In addition, according to different situations between the source and target domains, we can further categorize the *transductive transfer learning* setting into two cases.

- a. The feature spaces between the source and target domains are different,  $\mathcal{X}_S \neq \mathcal{X}_T$ .
- b. The feature spaces between domains are the same,  $\mathcal{X}_S = \mathcal{X}_T$ , but the marginal probability

distributions of the input data are different,  $P(X_S) \neq P(X_T)$ .

The latter case of the *transductive transfer learning* setting is related to domain adaptation for knowledge transfer in text classification [23] and sample selection bias [24] or covariate shift [25], whose assumptions are similar.

3. Finally, in the *unsupervised transfer learning* setting, similar to *inductive transfer learning* setting, the target task is different from but related to the source task. However, the *unsupervised transfer learning* focus on solving unsupervised learning tasks in the target domain, such as clustering, dimensionality reduction, and density estimation [26], [27]. In this case, there are no labeled data available in both source and target domains in training.

The relationship between the different settings of transfer learning and the related areas are summarized in Table 2 and Fig. 2.

Approaches to transfer learning in the above three different settings can be summarized into four cases based on "What to transfer." Table 3 shows these four cases and brief description. The first context can be referred to as instance-based transfer learning (or instance transfer) approach [6], [28], [29], [30], [31], [24], [32], [33], [34], [35], which assumes that certain parts of the data in the source domain can be reused for learning in the target domain by *reweighting*. Instance reweighting and importance sampling are two major techniques in this context.

A second case can be referred to as feature-representation-transfer approach [22], [36], [37], [38], [39], [8], [40], [41], [42], [43], [44]. The intuitive idea behind this case is to learn a "good" feature representation for the target domain. In this case, the knowledge used to transfer across domains is encoded into the learned feature representation. With the new feature representation, the performance of the target task is expected to improve significantly.

A third case can be referred to as parameter-transfer approach [45], [46], [47], [48], [49], which assumes that the source tasks and the target tasks share some parameters or prior distributions of the hyperparameters of the models. The transferred knowledge is encoded into the shared parameters or priors. Thus, by discovering the shared parameters or priors, knowledge can be transferred across tasks.

Finally, the last case can be referred to as the relational-knowledge-transfer problem [50], which deals with transfer learning for relational domains. The basic assumption

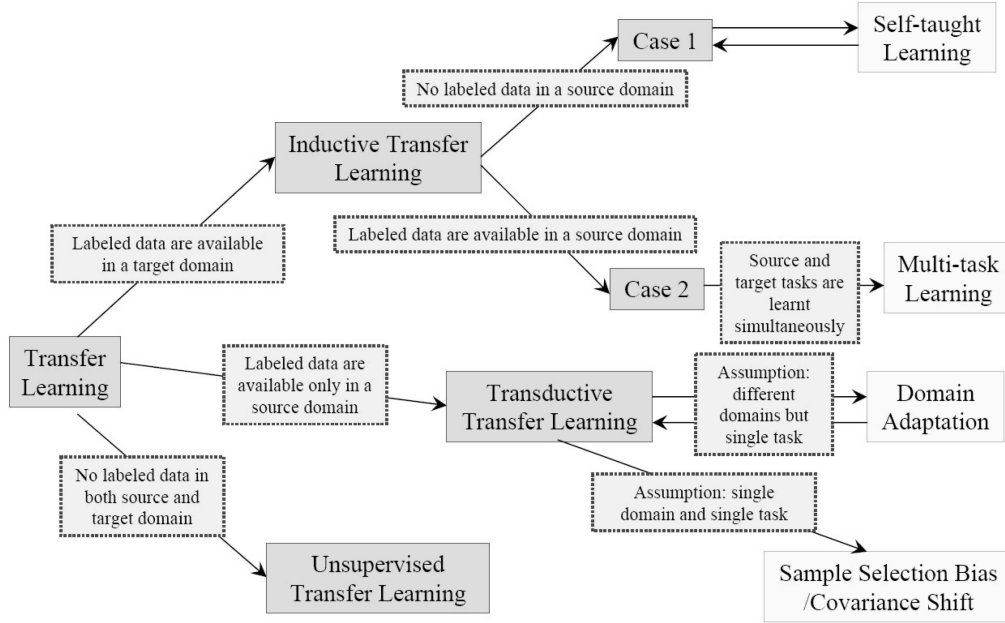


Fig. 2. An overview of different settings of transfer.

behind this context is that some relationship among the data in the source and target domains is similar. Thus, the knowledge to be transferred is the relationship among the data. Recently, statistical relational learning techniques dominate this context [51], [52].

Table 4 shows the cases where the different approaches are used for each transfer learning setting. We can see that the *inductive transfer learning* setting has been studied in many research works, while the *unsupervised transfer learning* setting is a relatively new research topic and only studied in the context of the *feature-representation-transfer* case. In addition, the *feature-representation-transfer* problem has been proposed to all three settings of transfer learning. However, the *parameter-transfer* and the *relational-knowledge-transfer* approach are only studied in the *inductive transfer learning* setting, which we discuss in detail below.

### 3 INDUCTIVE TRANSFER LEARNING

**Definition 2 (Inductive Transfer Learning).** Given a source domain  $\mathcal{D}_S$  and a learning task  $\mathcal{T}_S$ , a target domain  $\mathcal{D}_T$  and a learning task  $\mathcal{T}_T$ , inductive transfer learning aims to help improve the learning of the target predictive function

$f_T(\cdot)$  in  $\mathcal{D}_T$  using the knowledge in  $\mathcal{D}_S$  and  $\mathcal{T}_S$ , where  $\mathcal{T}_S \neq \mathcal{T}_T$ .

Based on the above definition of the *inductive transfer learning* setting, a few labeled data in the target domain are required as the training data to *induce* the target predictive function. As mentioned in Section 2.3, this setting has two cases: 1) labeled data in the source domain are available and 2) labeled data in the source domain are unavailable while unlabeled data in the source domain are available. Most transfer learning approaches in this setting focus on the former case.

#### 3.1 Transferring Knowledge of 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. [6] 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

TABLE 3  
Different Approaches to Transfer Learning

Transfer Learning Approaches	Brief Description
<i>Instance-transfer</i>	To re-weight some labeled data in the source domain for use in the target domain [6], [28], [29], [30], [31], [24], [32], [33], [34], [35].
<i>Feature-representation-transfer</i>	Find a “good” feature representation that reduces difference between the source and the target domains and the error of classification and regression models [22], [36], [37], [38], [39], [8], [40], [41], [42], [43], [44].
<i>Parameter-transfer</i>	Discover shared parameters or priors between the source domain and target domain models, which can benefit for transfer learning [45], [46], [47], [48], [49].
<i>Relational-knowledge-transfer</i>	Build mapping of relational knowledge between the source domain and the target domains. Both domains are relational domains and i.i.d assumption is relaxed in each domain [50], [51], [52].

TABLE 4  
Different Approaches Used in Different Settings

	Inductive Transfer Learning	Transductive Transfer Learning	Unsupervised Transfer Learning
<i>Instance-transfer</i>	✓	✓	
<i>Feature-representation-transfer</i>	✓	✓	✓
<i>Parameter-transfer</i>	✓		
<i>Relational-knowledge-transfer</i>	✓		

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 the target domains, 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. Theoretical analysis of *TrAdaBoost* is also given in [6].

Jiang and Zhai [30] proposed a heuristic method to remove “misleading” training examples from the source domain based on the difference between conditional probabilities  $P(y_T|x_T)$  and  $P(y_S|x_S)$ . Liao et al. [31] proposed a new active learning method to select the unlabeled data in a target domain to be labeled with the help of the source domain data. Wu and Dietterich [53] integrated the source domain (auxiliary) data an Support Vector Machine (SVM) framework for improving the classification performance.

### 3.2 Transferring Knowledge of 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 [40]. If no labeled data in the source domain are available, unsupervised learning methods are proposed to construct the feature representation.

#### 3.2.1 Supervised Feature Construction

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. [40] 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:

$$\begin{aligned} \arg \min_{A,U} \quad & \sum_{t \in \{T,S\}} \sum_{i=1}^{n_t} L(y_{t_i}, \langle a_t, U^T x_{t_i} \rangle) + \gamma \|A\|_{2,1}^2 \\ \text{s.t.} \quad & U \in \mathbf{O}^d. \end{aligned} \quad (1)$$

In this equation,  $S$  and  $T$  denote the tasks in the source domain and target domain, respectively.  $A = [a_S, a_T] \in R^{d \times 2}$  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  $\|A\|_{r,p} := (\sum_{i=1}^d \|a^i\|_r^p)^{\frac{1}{p}}$ . The optimization problem (1) 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. [41] proposed a spectral regularization framework on matrices for multitask structure learning.

Lee et al. [42] proposed a convex optimization algorithm for simultaneously learning metapriors and feature weights from an ensemble of related prediction tasks. The metapriors can be transferred among different tasks. Jebara [43] proposed to select features for multitask learning with SVMs. Rückert and Kramer [54] designed a kernel-based approach to inductive transfer, which aims at finding a suitable kernel for the target data.

#### 3.2.2 Unsupervised Feature Construction

In [22], Raina et al. proposed to apply sparse coding [55], which is an unsupervised feature construction method, for learning *higher level* features for transfer learning. The basic idea of this approach consists of two steps. In the first step, higher level basis vectors  $b = \{b_1, b_2, \dots, b_s\}$  are learned on the source domain data by solving the optimization problem (2) as shown as follows:

$$\begin{aligned} \min_{a,b} \sum_i \left\| x_{S_i} - \sum_j a_{S_i}^j b_j \right\|_2^2 + \beta \|a_{S_i}\|_1 \\ \text{s.t.} \quad \|b_j\|_2 \leq 1, \quad \forall j \in 1, \dots, s. \end{aligned} \quad (2)$$

In this equation,  $a_{S_i}^j$  is a new representation of basis  $b_j$  for input  $x_{S_i}$  and  $\beta$  is a coefficient to balance the feature construction term and the regularization term. After learning the basis vectors  $b$ , in the second step, an optimization algorithm (3) is applied on the target-domain data to learn *higher level* features based on the basis vectors  $b$ .

$$a_{T_i}^* = \arg \min_{a_{T_i}} \left\| x_{T_i} - \sum_j a_{T_i}^j b_j \right\|_2^2 + \beta \|a_{T_i}\|_1. \quad (3)$$

Finally, discriminative algorithms can be applied to  $\{a_{T_i}^*\}'s$  with corresponding labels to train classification or regression models for use in the target domain. One drawback of this method is that the so-called higher level basis vectors learned on the source domain in the optimization problem (2) may not be suitable for use in the target domain.

Recently, manifold learning methods have been adapted for transfer learning. In [44], Wang and Mahadevan proposed a Procrustes analysis-based approach to manifold alignment without correspondences, which can be used to transfer the knowledge across domains via the aligned manifolds.

### 3.3 Transferring Knowledge of 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 [45] 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. [46] 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. [47] 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 [48] 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*,

$$w_S = w_0 + v_S \text{ and } w_T = w_0 + v_T,$$

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  $f_t = w_t \cdot x$  to be a hyperplane for task  $t$ , an extension of SVMs to multitask learning case can be written as the following:

$$\begin{aligned} & \min_{w_0, v_t, \xi_{t_i}} J(w_0, v_t, \xi_{t_i}) \\ & = \sum_{t \in \{S, T\}} \sum_{i=1}^{n_t} \xi_{t_i} + \frac{\lambda_1}{2} \sum_{t \in \{S, T\}} \|v_t\|^2 + \lambda_2 \|w_0\|^2 \quad (4) \\ & \text{s.t. } y_{t_i}(w_0 + v_t) \cdot x_{t_i} \geq 1 - \xi_{t_i}, \\ & \xi_{t_i} \geq 0, \quad i \in \{1, 2, \dots, n_t\} \text{ and } t \in \{S, T\}. \end{aligned}$$

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

Several researchers have pursued the parameter-transfer approach further. Gao et al. [49] proposed a locally weighted ensemble learning framework to combine multiple models for transfer learning, where the weights are dynamically assigned according to a model's predictive power on each test example in the target domain.

### 3.4 Transferring 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.

Mihalkova et al. [50] proposed an algorithm *TAMAR* that transfers relational knowledge with Markov Logic Networks (MLNs) across relational domains. MLNs [56] is a powerful formalism, which combines the compact expressiveness of first-order logic with flexibility of probability, for statistical relational learning. In MLNs, entities in a relational domain are represented by predicates and their relationships are represented in first-order logic. *TAMAR* is motivated by the fact that if two domains are related to each other, there may exist mappings to connect entities and their relationships from a source domain to a target domain. For example, a professor can be considered as playing a similar role in an academic domain as a manager in an industrial management domain. In addition, the relationship between a professor and his or her students is similar to the relationship between a manager and his or her workers. Thus, there may exist a mapping from professor to manager and a mapping from the professor-student relationship to the manager-worker relationship. In this vein, *TAMAR* tries to use an MLN learned for a source domain to aid in the learning of an MLN for a target domain. Basically, *TAMAR* is a two-stage algorithm. In the first step, a mapping is constructed from a source MLN to the target domain based on weighted pseudo log-likelihood measure (WPLL). In the second step, a revision is done for the mapped structure in the target domain through the *FORTE* algorithm [57], which is an inductive logic programming (ILP) algorithm for revising first-order theories. The revised MLN can be used as a relational model for inference or reasoning in the target domain.

In the AAAI-2008 workshop on transfer learning for complex tasks,<sup>4</sup> Mihalkova and Mooney [51] extended

4. <http://www.cs.utexas.edu/~mtaylor/AAAI08TL/>.

TAMAR to the single-entity-centered setting of transfer learning, where only one entity in a target domain is available. Davis and Domingos [52] proposed an approach to transferring relational knowledge based on a form of second-order Markov logic. The basic idea of the algorithm is to discover structural regularities in the source domain in the form of Markov logic formulas with predicate variables, by instantiating these formulas with predicates from the target domain.

#### 4 TRANSDUCTIVE TRANSFER LEARNING

The term *transductive transfer learning* was first proposed by Arnold et al. [58], where they required that the source and target tasks be the same, although the domains may be different. On top of these conditions, they further required that all unlabeled data in the target domain are available at training time, but we believe that this condition can be relaxed; instead, in our definition of the *transductive transfer learning* setting, we only require that *part* of the unlabeled target data be seen at training time in order to obtain the marginal probability for the target data.

Note that the word “transductive” is used with several meanings. In the traditional machine learning setting, *transductive learning* [59] refers to the situation where all test data are required to be seen at training time, and that the learned model cannot be reused for future data. Thus, when some new test data arrive, they must be classified together with all existing data. In our categorization of transfer learning, in contrast, we use the term *transductive* to emphasize the concept that in this type of transfer learning, the tasks must be the same and there must be some unlabeled data available in the target domain.

**Definition 3 (Transductive Transfer Learning).** *Given a source domain  $\mathcal{D}_S$  and a corresponding learning task  $\mathcal{T}_S$ , a target domain  $\mathcal{D}_T$  and a corresponding learning task  $\mathcal{T}_T$ , transductive transfer learning aims to improve the learning of the target predictive function  $f_T(\cdot)$  in  $\mathcal{D}_T$  using the knowledge in  $\mathcal{D}_S$  and  $\mathcal{T}_S$ , where  $\mathcal{D}_S \neq \mathcal{D}_T$  and  $\mathcal{T}_S = \mathcal{T}_T$ . In addition, some unlabeled target-domain data must be available at training time.*

This definition covers the work of Arnold et al. [58], since the latter considered *domain adaptation*, where the difference lies between the marginal probability distributions of source and target data; i.e., the tasks are the same but the domains are different.

Similar to the traditional transductive learning setting, which aims to make the best use of the unlabeled test data for learning, in our classification scheme under transductive transfer learning, we also assume that some target-domain unlabeled data be given. In the above definition of transductive transfer learning, the source and target tasks are the same, which implies that one can adapt the predictive function learned in the source domain for use in the target domain through some unlabeled target-domain data. As mentioned in Section 2.3, this setting can be split to two cases: 1) The feature spaces between the source and target domains are different,  $\mathcal{X}_S \neq \mathcal{X}_T$ , and 2) the feature spaces between domains are the same,  $\mathcal{X}_S = \mathcal{X}_T$ , but the marginal probability distributions of the input data are different,  $P(X_S) \neq P(X_T)$ . This is similar to the requirements in domain adaptation and sample selection bias.

Most approaches described in the following sections are related to case 2 above.

##### 4.1 Transferring the Knowledge of Instances

Most instance-transfer approaches to the *transductive transfer learning* setting are motivated by importance sampling. To see how importance-sampling-based methods may help in this setting, we first review the problem of empirical risk minimization (ERM) [60]. In general, we might want to learn the optimal parameters  $\theta^*$  of the model by minimizing the expected risk,

$$\theta^* = \arg \min_{\theta \in \Theta} \mathbb{E}_{(x,y) \in P} [l(x, y, \theta)],$$

where  $l(x, y, \theta)$  is a loss function that depends on the parameter  $\theta$ . However, since it is hard to estimate the probability distribution  $P$ , we choose to minimize the ERM instead,

$$\theta^* = \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^n [l(x_i, y_i, \theta)],$$

where  $n$  is size of the training data.

In the *transductive transfer learning* setting, we want to learn an optimal model for the target domain by minimizing the expected risk,

$$\theta^* = \arg \min_{\theta \in \Theta} \sum_{(x,y) \in D_T} P(D_T) l(x, y, \theta).$$

However, since no labeled data in the target domain are observed in training data, we have to learn a model from the source domain data instead. If  $P(D_S) = P(D_T)$ , then we may simply learn the model by solving the following optimization problem for use in the target domain,

$$\theta^* = \arg \min_{\theta \in \Theta} \sum_{(x,y) \in D_S} P(D_S) l(x, y, \theta).$$

Otherwise, when  $P(D_S) \neq P(D_T)$ , we need to modify the above optimization problem to learn a model with high generalization ability for the target domain, as follows:

$$\begin{aligned} \theta^* &= \arg \min_{\theta \in \Theta} \sum_{(x,y) \in D_S} \frac{P(D_T)}{P(D_S)} P(D_S) l(x, y, \theta) \\ &\approx \arg \min_{\theta \in \Theta} \sum_{i=1}^{n_S} \frac{P_T(x_{T_i}, y_{T_i})}{P_S(x_{S_i}, y_{S_i})} l(x_{S_i}, y_{S_i}, \theta). \end{aligned} \quad (5)$$

Therefore, by adding different penalty values to each instance  $(x_{S_i}, y_{S_i})$  with the corresponding weight  $\frac{P_T(x_{T_i}, y_{T_i})}{P_S(x_{S_i}, y_{S_i})}$ , we can learn a precise model for the target domain. Furthermore, since  $P(Y_T|X_T) = P(Y_S|X_S)$ . Thus, the difference between  $P(D_S)$  and  $P(D_T)$  is caused by  $P(X_S)$  and  $P(X_T)$  and

$$\frac{P_T(x_{T_i}, y_{T_i})}{P_S(x_{S_i}, y_{S_i})} = \frac{P(x_{S_i})}{P(x_{T_i})}.$$

If we can estimate  $\frac{P(x_{S_i})}{P(x_{T_i})}$  for each instance, we can solve the *transductive transfer learning* problems.

There exist various ways to estimate  $\frac{P(x_{S_i})}{P(x_{T_i})}$ . Zadrozny [24] proposed to estimate the terms  $P(x_{S_i})$  and  $P(x_{T_i})$  independently by constructing simple classification problems.



Fan et al. [35] further analyzed the problems by using various classifiers to estimate the probability ratio. Huang et al. [32] proposed a kernel-mean matching (KMM) algorithm to learn  $\frac{P(x_{S_i})}{P(x_{T_i})}$  directly by matching the means between the source domain data and the target domain data in a reproducing-kernel Hilbert space (RKHS). KMM can be rewritten as the following quadratic programming (QP) optimization problem.

$$\begin{aligned} \min_{\beta} \quad & \frac{1}{2} \beta^T K \beta - \kappa^T \beta \\ \text{s.t.} \quad & \beta_i \in [0, B] \text{ and } \left| \sum_{i=1}^{n_S} \beta_i - n_S \right| \leq n_S \epsilon, \end{aligned} \quad (6)$$

where

$$K = \begin{bmatrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{bmatrix}$$

and  $K_{ij} = k(x_i, x_j)$ .  $K_{S,S}$  and  $K_{T,T}$  are kernel matrices for the source domain data and the target domain data, respectively.  $\kappa_i = \frac{n_S}{n_T} \sum_{j=1}^{n_T} k(x_i, x_{T_j})$ , where  $x_i \in X_S \cup X_T$ , while  $x_{T_j} \in X_T$ .

It can be proved that  $\beta_i = \frac{P(x_{S_i})}{P(x_{T_i})}$  [32]. An advantage of using KMM is that it can avoid performing density estimation of either  $P(x_{S_i})$  or  $P(x_{T_i})$ , which is difficult when the size of the data set is small. Sugiyama et al. [34] proposed an algorithm known as Kullback-Leibler Importance Estimation Procedure (KLIEP) to estimate  $\frac{P(x_{S_i})}{P(x_{T_i})}$  directly, based on the minimization of the Kullback-Leibler divergence. can be integrated with cross-validation to perform model selection automatically in two steps: 1) estimating the weights of the source domain data and 2) training models on the reweighted data. Bickel et al. [33] combined the two steps in a unified framework by deriving a kernel-logistic regression classifier. Besides sample reweighting techniques, Dai et al. [28] extended a traditional Naive Bayesian classifier for the transductive transfer learning problems. For more information on importance sampling and reweighting methods for covariate shift or sample selection bias, readers can refer to a recently published book [29] by Quionero-Candela et al. One can also consult a tutorial on Sample Selection Bias by Fan and Sugiyama in ICDM-08.<sup>5</sup>

## 4.2 Transferring Knowledge of Feature Representations

Most feature-representation-transfer approaches to the transductive transfer learning setting are under unsupervised learning frameworks. Blitzer et al. [38] proposed a structural correspondence learning (SCL) algorithm, which extends [37], to make use of the unlabeled data from the target domain to extract some relevant features that may reduce the difference between the domains. The first step of SCL is to define a set of *pivot* features<sup>6</sup> (the number of *pivot* feature is denoted by  $m$ ) on the unlabeled data from both

domains. Then, SCL removes these *pivot* features from the data and treats each *pivot* feature as a new label vector. The  $m$  classification problems can be constructed. By assuming each problem can be solved by linear classifier, which is shown as follows:

$$f_l(x) = \text{sgn}(w_l^T \cdot x), \quad l = 1, \dots, m.$$

SCL can learn a matrix  $W = [w_1 w_2 \dots w_m]$  of parameters. In the third step, singular value decomposition (SVD) is applied to matrix  $W = [w_1 w_2 \dots w_m]$ . Let  $W = UDV^T$ , then  $\theta = U_{[1:h,:]}^T$  ( $h$  is the number of the shared features) is the matrix (linear mapping) whose rows are the top left singular vectors of  $W$ . Finally, standard discriminative algorithms can be applied to the augmented feature vector to build models. The augmented feature vector contains all the original feature  $x_i$  appended with the new shared features  $\theta x_i$ . As mentioned in [38], if the *pivot* features are well designed, then the learned mapping  $\theta$  encodes the correspondence between the features from the different domains. Although Ben-David et al. [61] showed experimentally that SCL can reduce the difference between domains; how to select the *pivot* features is difficult and domain dependent. In [38], Blitzer et al. used a heuristic method to select pivot features for natural language processing (NLP) problems, such as tagging of sentences. In their follow-up work, the researchers proposed to use Mutual Information (MI) to choose the pivot features instead of using more heuristic criteria [8]. MI-SCL tries to find some pivot features that have high dependence on the labels in the source domain.

Transfer learning in the NLP domain is sometimes referred to as domain adaptation. In this area, Daumé [39] proposed a kernel-mapping function for NLP problems, which maps the data from both source and target domains to a high-dimensional feature space, where standard discriminative learning methods are used to train the classifiers. However, the constructed kernel-mapping function is domain knowledge driven. It is not easy to generalize the kernel mapping to other areas or applications. Blitzer et al. [62] analyzed the uniform convergence bounds for algorithms that minimized a convex combination of source and target empirical risks.

In [36], Dai et al. proposed a coclustering-based algorithm to propagate the label information across different domains. In [63], Xing et al. proposed a novel algorithm known as *bridged refinement* to correct the labels predicted by a shift-unaware classifier toward a target distribution and take the mixture distribution of the training and test data as a bridge to better transfer from the training data to the test data. In [64], Ling et al. proposed a spectral classification framework for cross-domain transfer learning problem, where the objective function is introduced to seek consistency between the in-domain supervision and the out-of-domain intrinsic structure. In [65], Xue et al. proposed a cross-domain text classification algorithm that extended the traditional probabilistic latent semantic analysis (PLSA) algorithm to integrate labeled and unlabeled data from different but related domains, into a unified probabilistic model. The new model is called Topic-bridged PLSA, or TPLSA.

Transfer learning via dimensionality reduction was recently proposed by Pan et al. [66]. In this work, Pan et al. exploited the Maximum Mean Discrepancy Embedding

<sup>5</sup>. Tutorial slides can be found at <http://www.cs.columbia.edu/~fan/PPT/ICDM08SampleBias.ppt>.

<sup>6</sup>. The *pivot* features are domain specific and depend on prior knowledge.

(MMDE) method, originally designed for dimensionality reduction, to learn a low-dimensional space to reduce the difference of distributions between different domains for transductive transfer learning. However, MMDE may suffer from its computational burden. Thus, in [67], Pan et al. further proposed an efficient feature extraction algorithm, known as Transfer Component Analysis (TCA) to overcome the drawback of MMDE.

## 5 UNSUPERVISED TRANSFER LEARNING

**Definition 4 (Unsupervised Transfer Learning).** *Given a source domain  $\mathcal{D}_S$  with a learning task  $\mathcal{T}_S$ , a target domain  $\mathcal{D}_T$  and a corresponding learning task  $\mathcal{T}_T$ , unsupervised transfer learning aims to help improve the learning of the target predictive function  $f_T(\cdot)$ <sup>7</sup> in  $\mathcal{D}_T$  using the knowledge in  $\mathcal{D}_S$  and  $\mathcal{T}_S$ , where  $\mathcal{T}_S \neq \mathcal{T}_T$  and  $\mathcal{Y}_S$  and  $\mathcal{Y}_T$  are not observable.*

Based on the definition of the *unsupervised transfer learning* setting, no labeled data are observed in the source and target domains in training. So far, there is little research work on this setting. Recently, *Self-taught clustering* (STC) [26] and transferred discriminative analysis (TDA) [27] algorithms are proposed to transfer clustering and transfer dimensionality reduction problems, respectively.

### 5.1 Transferring Knowledge of Feature Representations

Dai et al. [26] studied a new case of clustering problems, known as *self-taught clustering*. *Self-taught clustering* is an instance of *unsupervised transfer learning*, which aims at clustering a small collection of unlabeled data in the target domain with the help of a large amount of unlabeled data in the source domain. STC tries to learn a common feature space across domains, which helps in clustering in the target domain. The objective function of STC is shown as follows:

$$J(\tilde{X}_T, \tilde{X}_S, \tilde{Z}) = I(X_T, Z) - I(\tilde{X}_T, \tilde{Z}) + \lambda [I(X_S, Z) - I(\tilde{X}_S, \tilde{Z})], \quad (7)$$

where  $X_S$  and  $X_T$  are the source and target domain data, respectively.  $Z$  is a shared feature space by  $X_S$  and  $X_T$ , and  $I(\cdot, \cdot)$  is the mutual information between two random variables. Suppose that there exist three clustering functions  $C_{X_T}: X_T \rightarrow \tilde{X}_T$ ,  $C_{X_S}: X_S \rightarrow \tilde{X}_S$ , and  $C_Z: Z \rightarrow \tilde{Z}$ , where  $\tilde{X}_T$ ,  $\tilde{X}_S$ , and  $\tilde{Z}$  are corresponding clusters of  $X_T$ ,  $X_S$ , and  $Z$ , respectively. The goal of STC is to learn  $\tilde{X}_T$  by solving the optimization problem (7):

$$\arg \min_{\tilde{X}_T, \tilde{X}_S, \tilde{Z}} J(\tilde{X}_T, \tilde{X}_S, \tilde{Z}). \quad (8)$$

An iterative algorithm for solving the optimization function (8) was given in [26].

Similarly, Wang et al. [27] proposed a TDA algorithm to solve the *transfer dimensionality reduction* problem. TDA first applies clustering methods to generate pseudoclass labels for the target unlabeled data. It then applies dimensionality reduction methods to the target data and labeled source

data to reduce the dimensions. These two steps run iteratively to find the best subspace for the target data.

## 6 TRANSFER BOUNDS AND NEGATIVE TRANSFER

An important issue is to recognize the limit of the power of transfer learning. In [68], Mahmud and Ray analyzed the case of transfer learning using Kolmogorov complexity, where some theoretical bounds are proved. In particular, the authors used conditional Kolmogorov complexity to measure relatedness between tasks and transfer the “right” amount of information in a sequential transfer learning task under a Bayesian framework.

Recently, Eaton et al. [69] proposed a novel graph-based method for knowledge transfer, where the relationships between source tasks are modeled by embedding the set of learned source models in a graph using transferability as the metric. Transferring to a new task proceeds by mapping the problem into the graph and then learning a function on this graph that automatically determines the parameters to transfer to the new learning task.

Negative transfer happens when the source domain data and task contribute to the reduced performance of learning in the target domain. Despite the fact that how to avoid negative transfer is a very important issue, little research work has been published on this topic. Rosenstein et al. [70] empirically showed that if two tasks are too dissimilar, then brute-force transfer may hurt the performance of the target task. Some works have been exploited to analyze relatedness among tasks and task clustering techniques, such as [71], [72], which may help provide guidance on how to avoid negative transfer automatically. Bakker and Heskes [72] adopted a Bayesian approach in which some of the model parameters are shared for all tasks and others more loosely connected through a joint prior distribution that can be learned from the data. Thus, the data are clustered based on the task parameters, where tasks in the same cluster are supposed to be related to each other. Argyriou et al. [73] considered situations in which the learning tasks can be divided into groups. Tasks within each group are related by sharing a low-dimensional representation, which differs among different groups. As a result, tasks within a group can find it easier to transfer useful knowledge.

## 7 APPLICATIONS OF TRANSFER LEARNING

Recently, transfer learning techniques have been applied successfully in many real-world applications. Raina et al. [74] and Dai et al. [36], [28] proposed to use transfer learning techniques to learn text data across domains, respectively. Blitzer et al. [38] proposed to use SCL for solving NLP problems. An extension of SCL was proposed in [8] for solving sentiment classification problems. Wu and Dietterich [53] proposed to use both inadequate target domain data and plenty of low quality source domain data for image classification problems. Arnold et al. [58] proposed to use *transductive transfer learning* methods to solve name-entity recognition problems. In [75], [76], [77], [78], [79], transfer learning techniques are proposed to extract knowledge from WiFi localization models across time periods, space, and mobile devices, to benefit WiFi

7. In unsupervised transfer learning, the predicted labels are latent variables, such as clusters or reduced dimensions.

localization tasks in other settings. Zhuo et al. [80] studied how to transfer domain knowledge to learn relational action models across domains in automated planning.

In [81], Raykar et al. proposed a novel Bayesian multiple-instance learning algorithm, which can automatically identify the relevant feature subset and use inductive transfer for learning multiple, but conceptually related, classifiers, for computer aided design (CAD). In [82], Ling et al. proposed an information-theoretic approach for transfer learning to address the *cross-language classification problem* for translating webpages from English to Chinese. The approach addressed the problem when there are plenty of labeled English text data whereas there are only a small number of labeled Chinese text documents. Transfer learning across the two feature spaces are achieved by designing a suitable mapping function as a bridge.

So far, there are at least two international competitions based on transfer learning, which made available some much needed public data. In the ECML/PKDD-2006 discovery challenge,<sup>8</sup> the task was to handle personalized spam filtering and generalization across related learning tasks. For training a spam-filtering system, we need to collect a lot of e-mails from a group of users with corresponding labels: *spam* or *not spam*, and train a classifier based on these data. For a new e-mail user, we might want to adapt the learned model for the user. The challenge is that the distributions of emails for the first set of users and the new user are different. Thus, this problem can be modeled as an inductive transfer learning problem, which aims to adapt an old spam-filtering model to a new situation with fewer training data and less training time.

A second data set was made available through the ICDM-2007 Contest, in which a task was to estimate a WiFi client's indoor locations using the WiFi signal data obtained over different periods of time [83]. Since the values of WiFi signal strength may be a function of time, space, and devices, distributions of WiFi data over different time periods may be very different. Thus, transfer learning must be designed to reduce the data relabeling effort.

**Data sets for transfer learning.** So far, several data sets have been published for transfer learning research. We denote the text mining data sets, Email spam-filtering data set, the WiFi localization over time periods data set, and the Sentiment classification data set by **Text**, **E-mail**, **WiFi**, and **Sen**, respectively.

1. **Text.** Three data sets, 20 Newsgroups, SRAA, and Reuters-21578,<sup>9</sup> have been preprocessed for a transfer learning setting by some researchers. The data in these data sets are categorized to a hierarchical structure. Data from different subcategories under the same parent category are considered to be from different but related domains. The task is to predict the labels of the parent category.
2. **E-mail.** This data set is provided by the 2006 ECML/PKDD discovery challenge.
3. **WiFi.** This data set is provided by the ICDM-2007 Contest.<sup>10</sup> The data were collected inside a building

for localization around  $145.5 \times 37.5 \text{ m}^2$  in two different time periods.

4. **Sen.** This data set was first used in [8].<sup>11</sup> This data set contains product reviews downloaded from Amazon.com from four product types (domains): Kitchen, Books, DVDs, and Electronics. Each domain has several thousand reviews, but the exact number varies by domain. Reviews contain star ratings (1-5 stars).

**Empirical evaluation.** To show how much benefit transfer learning methods can bring as compared to traditional learning methods, researchers have used some public data sets. We show a list taken from some published transfer learning papers in Table 5. In [6], [84], [49], the authors used the 20 Newsgroups data<sup>12</sup> as one of the evaluation data sets. Due to the differences in the pre-processing steps of the algorithms by different researchers, it is hard to compare the proposed methods directly. Thus, we denote them by 20-Newsgroups<sub>1</sub>, 20-Newsgroups<sub>2</sub>, and 20-Newsgroups<sub>3</sub>, respectively, and show the comparison results between the proposed transfer learning methods and nontransfer learning methods in the table.

On the 20 Newsgroups<sub>1</sub> data, Dai et al. [6] showed the comparison experiments between standard SVM and the proposed TrAdaBoost algorithm. On 20 Newsgroups<sub>2</sub>, Shi et al. [84] applied an active learning algorithm to select important instances for transfer learning (AcTraK) with TrAdaBoost and standard SVM. Gao et al. [49] evaluated their proposed locally weighted ensemble learning algorithms, pLWE and LWE, on the 20 Newsgroups<sub>3</sub>, compared to SVM and Logistic Regression (LR).

In addition, in the table, we also show the comparison results on the sentiment classification data set reported in [8]. On this data set, SGD denotes the stochastic gradient-descent algorithm with Huber loss, SCL represents a linear predictor on the new representations learned by Structural Correspondence Learning algorithm, and SCL-MI is an extension of SCL by applying Mutual Information to select the pivot features for the SCL algorithm.

Finally, on the WiFi localization data set, we show the comparison results reported in [67], where the baseline is a regularized least square regression model (RLSR), which is a standard regression model, and KPCA, which represents to apply RLSR on the new representations of the data learned by Kernel Principle Component Analysis. The compared transfer learning methods include KMM and the proposed algorithm, TCA. For more detail about the experimental results, the readers may refer to the reference papers showed in the table. From these comparison results, we can find that the transfer learning methods designed appropriately for real-world applications can indeed improve the performance significantly compared to the nontransfer learning methods.

**Toolboxes for transfer learning.** Researchers at UC Berkeley provided a MATLAB toolkit for transfer learning.<sup>13</sup> The toolkit contains algorithms and benchmark data sets for transfer learning. In addition, it provides a standard

8. <http://www.ecmlpkdd2006.org/challenge.html>.

9. [http://apex.sjtu.edu.cn/apex\\_wiki/dwyak](http://apex.sjtu.edu.cn/apex_wiki/dwyak).

10. <http://www.cse.ust.hk/~qyang/ICDMDMC2007>.

11. <http://www.cis.upenn.edu/~mdredze/datasets/sentiment/>.

12. <http://people.csail.mit.edu/jrennie/20Newsgroups/>.

13. <http://multitask.cs.berkeley.edu/>.

TABLE 5  
Comparison between Transfer Learning and Nontransfer Learning Methods

Data Set (reference)	Source v.s. Target	Baselines		TL Methods	
20 Newsgroups <sub>1</sub> ([6]) ACC (unit: %)		<b>SVM</b>		<b>TrAdaBoost</b>	
	rec v.s. talk	87.3%		92.0%	
	rec v.s. sci	83.6%		90.3%	
	sci v.s. talk	82.3%		87.5%	
20 Newsgroups <sub>2</sub> ([84]) ACC (unit: %)		<b>SVM</b>		<b>TrAdaBoost</b>	<b>AcTraK</b>
	rec v.s. talk	60.2%		72.3%	75.4%
	rec v.s. sci	59.1%		67.4%	70.6%
	comp v.s. talk	53.6%		74.4%	80.9%
	comp v.s. sci	52.7%		57.3%	78.0%
	comp v.s. rec	49.1%		77.2%	82.1%
	sci v.s. talk	57.6%		71.3%	75.1%
20 Newsgroups <sub>3</sub> ([49]) ACC (unit: %)		<b>SVM</b>	<b>LR</b>	<b>pLWE</b>	<b>LWE</b>
	comp v.s. sci	71.18%	73.49%	78.72%	97.44%
	rec v.s. talk	68.24%	72.17%	72.17%	99.23%
	rec v.s. sci	78.16%	78.85%	88.45%	98.23%
	sci v.s. talk	75.77%	79.04%	83.30%	96.92%
	comp v.s. rec	81.56%	83.34%	91.93%	98.16%
	comp v.s. talk	93.89%	91.76%	96.64%	98.90%
Sentiment Classification ([8]) ACC (unit: %)		<b>SGD</b>		<b>SCL</b>	<b>SCL-MI</b>
	DVD v.s. book	72.8%		76.8%	79.7%
	electronics v.s. book	70.7%		75.4%	75.4%
	kitchen v.s. book	70.9%		66.1%	68.6%
	book v.s. DVD	77.2%		74.0%	75.8%
	electronics v.s. DVD	70.6%		74.3%	76.2%
	kitchen v.s. DVD	72.7%		75.4%	76.9%
	book v.s. electronics	70.8%		77.5%	75.9%
	DVD v.s. electronics	73.0%		74.1%	74.1%
	kitchen v.s. electronics	82.7%		83.7%	86.8%
	book v.s. kitchen	74.5%		78.7%	78.9%
	DVD v.s. kitchen	74.0%		79.4%	81.4%
electronics v.s. kitchen	84.0%		84.4%	85.9%	
WiFi Localization ([67]) AED (unit: meter)		<b>RLSR</b>	<b>PCA</b>	<b>KMM</b>	<b>TCA</b>
	Time A v.s. Time B	6.52	3.16	5.51	2.37

platform for developing and testing new algorithms for transfer learning.

### 7.1 Other Applications of Transfer Learning

Transfer learning has found many applications in sequential machine learning as well. For example, Kuhlmann and Stone [85] proposed a graph-based method for identifying previously encountered games, and applied this technique to automate domain mapping for value function transfer and speed up reinforcement learning on variants of previously played games. A new approach to transfer between entirely different feature spaces is proposed in *translated learning*, which is made possible by learning a mapping function for bridging features in two entirely different domains (images and text) [86]. Finally, Li et al. [87], [88] have applied transfer learning to collaborative filtering problems to solve the cold start and sparsity problems. In [87], Li et al. learned a shared rating-pattern mixture model, known as a Rating-Matrix Generative Model (RMGM), in terms of the latent user- and item-cluster variables. RMGM bridges multiple rating matrices from different domains by mapping the users and items in each rating matrix onto the shared latent user and item spaces in order to transfer useful knowledge. In [88], they applied coclustering algorithms on users and items in an

auxiliary rating matrix. They then constructed a cluster-level rating matrix known as a codebook. By assuming the target rating matrix (on movies) is related to the auxiliary one (on books), the target domain can be reconstructed by expanding the codebook, completing the knowledge transfer process.

## 8 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.

Furthermore, each of the approaches to transfer learning can be classified into four contexts based on “what to transfer” in learning. They include the instance-transfer approach, the feature-representation-transfer approach, the parameter-transfer approach, and the relational-knowledge-transfer approach, respectively. The former three contexts have an i.i.d. assumption on the data while the last context deals with transfer learning on relational data. Most of these approaches assume that the selected source domain is related to the target domain.

In the future, several important research issues need to be addressed. First, how to avoid negative transfer is an open problem. As mentioned in Section 6, many proposed transfer learning algorithms assume that the source and target domains are related to each other in some sense. However, if the assumption does not hold, negative transfer may happen, which may cause the learner to perform worse than no transferring at all. Thus, how to make sure that no negative transfer happens is a crucial issue in transfer learning. In order to avoid negative transfer learning, we need to first study transferability between source domains or tasks and target domains or tasks. Based on suitable transferability measures, we can then select relevant source domains or tasks to extract knowledge from for learning the target tasks. To define the transferability between domains and tasks, we also need to define the criteria to measure the similarity between domains or tasks. Based on the distance measures, we can then cluster domains or tasks, which may help measure transferability. A related issue is when an entire domain cannot be used for transfer learning, whether we can still transfer part of the domain for useful learning in the target domain.

In addition, most existing transfer learning algorithms so far focused on improving generalization across different distributions between source and target domains or tasks. In doing so, they assumed that the feature spaces between the source and target domains are the same. However, in many applications, we may wish to transfer knowledge across domains or tasks that have different feature spaces, and transfer from multiple such source domains. We refer to this type of transfer learning as *heterogeneous transfer learning*.

Finally, so far, transfer learning techniques have been mainly applied to small scale applications with a limited variety, such as sensor-network-based localization, text classification, and image classification problems. In the future, transfer learning techniques will be widely used to solve other challenging applications, such as video classification, social network analysis, and logical inference.

## ACKNOWLEDGMENTS

The authors thank the support of Hong Kong CERG Project 621307 and a grant from NEC China Lab.

## REFERENCES

- [1] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A.F.M. Ng, B. Liu, P.S. Yu, Z.-H. Zhou, M. Steinbach, D.J. Hand, and D. Steinberg, "Top 10 Algorithms in Data Mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1-37, 2008.
- [2] Q. Yang and X. Wu, "10 Challenging Problems in Data Mining Research," *Int'l J. Information Technology and Decision Making*, vol. 5, no. 4, pp. 597-604, 2006.
- [3] G.P.C. Fung, J.X. Yu, H. Lu, and P.S. Yu, "Text Classification without Negative Examples Revisited," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 1, pp. 6-20, Jan. 2006.
- [4] H. Al Mubaid and S.A. Umair, "A New Text Categorization Technique Using Distributional Clustering and Learning Logic," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 9, pp. 1156-1165, Sept. 2006.
- [5] K. Sarinapakorn and M. Kubat, "Combining Subclassifiers in Text Categorization: A DST-Based Solution and a Case Study," *IEEE Trans. Knowledge and Data Eng.*, vol. 19, no. 12, pp. 1638-1651, Dec. 2007.
- [6] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for Transfer Learning," *Proc. 24th Int'l Conf. Machine Learning*, pp. 193-200, June 2007.
- [7] S.J. Pan, V.W. Zheng, Q. Yang, and D.H. Hu, "Transfer Learning for WiFi-Based Indoor Localization," *Proc. Workshop Transfer Learning for Complex Task of the 23rd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*, July 2008.
- [8] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, Boom-Boxes and Blenders: Domain Adaptation for Sentiment Classification," *Proc. 45th Ann. Meeting of the Assoc. Computational Linguistics*, pp. 432-439, 2007.
- [9] J. Ramon, K. Driessens, and T. Croonenborghs, "Transfer Learning in Reinforcement Learning Problems through Partial Policy Recycling," *Proc. 18th European Conf. Machine Learning (ECML '07)*, pp. 699-707, 2007.
- [10] M.E. Taylor and P. Stone, "Cross-Domain Transfer for Reinforcement Learning," *Proc. 24th Int'l Conf. Machine Learning (ICML '07)*, pp. 879-886, 2007.
- [11] X. Yin, J. Han, J. Yang, and P.S. Yu, "Efficient Classification across Multiple Database Relations: A Crossmine Approach," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 6, pp. 770-783, June 2006.
- [12] L.I. Kuncheva and J.J. Rodríguez, "Classifier Ensembles with a Random Linear Oracle," *IEEE Trans. Knowledge and Data Eng.*, vol. 19, no. 4, pp. 500-508, Apr. 2007.
- [13] E. Baralis, S. Chiusano, and P. Garza, "A Lazy Approach to Associative Classification," *IEEE Trans. Knowledge and Data Eng.*, vol. 20, no. 2, pp. 156-171, Feb. 2008.
- [14] X. Zhu, "Semi-Supervised Learning Literature Survey," Technical Report 1530, Univ. of Wisconsin-Madison, 2006.
- [15] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell, "Text Classification from Labeled and Unlabeled Documents Using EM," *Machine Learning*, vol. 39, nos. 2/3, pp. 103-134, 2000.
- [16] A. Blum and T. Mitchell, "Combining Labeled and Unlabeled Data with Co-Training," *Proc. 11th Ann. Conf. Computational Learning Theory*, pp. 92-100, 1998.
- [17] T. Joachims, "Transductive Inference for Text Classification Using Support Vector Machines," *Proc. 16th Int'l Conf. Machine Learning*, pp. 825-830, 1999.
- [18] X. Zhu and X. Wu, "Class Noise Handling for Effective Cost-Sensitive Learning by Cost-Guided Iterative Classification Filtering," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 10, pp. 1435-1440, Oct. 2006.
- [19] Q. Yang, C. Ling, X. Chai, and R. Pan, "Test-Cost Sensitive Classification on Data with Missing Values," *IEEE Trans. Knowledge and Data Eng.*, vol. 18, no. 5, pp. 626-638, May 2006.
- [20] *Learning to Learn*. S. Thrun and L. Pratt, eds. Kluwer Academic Publishers, 1998.
- [21] R. Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41-75, 1997.
- [22] R. Raina, A. Battle, H. Lee, B. Packer, and A.Y. Ng, "Self-Taught Learning: Transfer Learning from Unlabeled Data," *Proc. 24th Int'l Conf. Machine Learning*, pp. 759-766, June 2007.
- [23] H. Daumé III and D. Marcu, "Domain Adaptation for Statistical Classifiers," *J. Artificial Intelligence Research*, vol. 26, pp. 101-126, 2006.
- [24] B. Zadrozny, "Learning and Evaluating Classifiers under Sample Selection Bias," *Proc. 21st Int'l Conf. Machine Learning*, July 2004.
- [25] H. Shimodaira, "Improving Predictive Inference under Covariate Shift by Weighting the Log-Likelihood Function," *J. Statistical Planning and Inference*, vol. 90, pp. 227-244, 2000.
- [26] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Self-Taught Clustering," *Proc. 25th Int'l Conf. Machine Learning*, pp. 200-207, July 2008.
- [27] Z. Wang, Y. Song, and C. Zhang, "Transferred Dimensionality Reduction," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML/PKDD '08)*, pp. 550-565, Sept. 2008.
- [28] W. Dai, G. Xue, Q. Yang, and Y. Yu, "Transferring Naive Bayes Classifiers for Text Classification," *Proc. 22nd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*, pp. 540-545, July 2007.
- [29] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N.D. Lawrence, *Dataset Shift in Machine Learning*. MIT Press, 2009.
- [30] J. Jiang and C. Zhai, "Instance Weighting for Domain Adaptation in NLP," *Proc. 45th Ann. Meeting of the Assoc. Computational Linguistics*, pp. 264-271, June 2007.

- [31] X. Liao, Y. Xue, and L. Carin, "Logistic Regression with an Auxiliary Data Source," *Proc. 21st Int'l Conf. Machine Learning*, pp. 505-512, Aug. 2005.
- [32] J. Huang, A. Smola, A. Gretton, K.M. Borgwardt, and B. Schölkopf, "Correcting Sample Selection Bias by Unlabeled Data," *Proc. 19th Ann. Conf. Neural Information Processing Systems*, 2007.
- [33] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative Learning for Differing Training and Test Distributions," *Proc. 24th Int'l Conf. Machine Learning*, pp. 81-88, 2007.
- [34] M. Sugiyama, S. Nakajima, H. Kashima, P.V. Buenau, and M. Kawanabe, "Direct Importance Estimation with Model Selection and its Application to Covariate Shift Adaptation," *Proc. 20th Ann. Conf. Neural Information Processing Systems*, Dec. 2008.
- [35] W. Fan, I. Davidson, B. Zadrozny, and P.S. Yu, "An Improved Categorization of Classifier's Sensitivity on Sample Selection Bias," *Proc. Fifth IEEE Int'l Conf. Data Mining*, 2005.
- [36] W. Dai, G. Xue, Q. Yang, and Y. Yu, "Co-Clustering Based Classification for Out-of-Domain Documents," *Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, Aug. 2007.
- [37] R.K. Ando and T. Zhang, "A High-Performance Semi-Supervised Learning Method for Text Chunking," *Proc. 43rd Ann. Meeting on Assoc. for Computational Linguistics*, pp. 1-9, 2005.
- [38] J. Blitzer, R. McDonald, and F. Pereira, "Domain Adaptation with Structural Correspondence Learning," *Proc. Conf. Empirical Methods in Natural Language*, pp. 120-128, July 2006.
- [39] H. Daumé III, "Frustratingly Easy Domain Adaptation," *Proc. 45th Ann. Meeting of the Assoc. Computational Linguistics*, pp. 256-263, June 2007.
- [40] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-Task Feature Learning," *Proc. 19th Ann. Conf. Neural Information Processing Systems*, pp. 41-48, Dec. 2007.
- [41] A. Argyriou, C.A. Micchelli, M. Pontil, and Y. Ying, "A Spectral Regularization Framework for Multi-Task Structure Learning," *Proc. 20th Ann. Conf. Neural Information Processing Systems*, pp. 25-32, 2008.
- [42] S.I. Lee, V. Chatalbashev, D. Vickrey, and D. Koller, "Learning a Meta-Level Prior for Feature Relevance from Multiple Related Tasks," *Proc. 24th Int'l Conf. Machine Learning*, pp. 489-496, July 2007.
- [43] T. Jebara, "Multi-Task Feature and Kernel Selection for SVMs," *Proc. 21st Int'l Conf. Machine Learning*, July 2004.
- [44] C. Wang and S. Mahadevan, "Manifold Alignment Using Procrustes Analysis," *Proc. 25th Int'l Conf. Machine Learning*, pp. 1120-1127, July 2008.
- [45] N.D. Lawrence and J.C. Platt, "Learning to Learn with the Informative Vector Machine," *Proc. 21st Int'l Conf. Machine Learning*, July 2004.
- [46] E. Bonilla, K.M. Chai, and C. Williams, "Multi-Task Gaussian Process Prediction," *Proc. 20th Ann. Conf. Neural Information Processing Systems*, pp. 153-160, 2008.
- [47] A. Schwaighofer, V. Tresp, and K. Yu, "Learning Gaussian Process Kernels via Hierarchical Bayes," *Proc. 17th Ann. Conf. Neural Information Processing Systems*, pp. 1209-1216, 2005.
- [48] T. Evgeniou and M. Pontil, "Regularized Multi-Task Learning," *Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 109-117, Aug. 2004.
- [49] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge Transfer via Multiple Model Local Structure Mapping," *Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 283-291, Aug. 2008.
- [50] L. Mihalkova, T. Huynh, and R.J. Mooney, "Mapping and Revising Markov Logic Networks for Transfer Learning," *Proc. 22nd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*, pp. 608-614, July 2007.
- [51] L. Mihalkova and R.J. Mooney, "Transfer Learning by Mapping with Minimal Target Data," *Proc. Assoc. for the Advancement of Artificial Intelligence (AAAI '08) Workshop Transfer Learning for Complex Tasks*, July 2008.
- [52] J. Davis and P. Domingos, "Deep Transfer via Second-Order Markov Logic," *Proc. Assoc. for the Advancement of Artificial Intelligence (AAAI '08) Workshop Transfer Learning for Complex Tasks*, July 2008.
- [53] P. Wu and T.G. Dietterich, "Improving SVM Accuracy by Training on Auxiliary Data Sources," *Proc. 21st Int'l Conf. Machine Learning*, July 2004.
- [54] U. Rückert and S. Kramer, "Kernel-Based Inductive Transfer," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML/PKDD '08)*, pp. 220-233, Sept. 2008.
- [55] H. Lee, A. Battle, R. Raina, and A.Y. Ng, "Efficient Sparse Coding Algorithms," *Proc. 19th Ann. Conf. Neural Information Processing Systems*, pp. 801-808, 2007.
- [56] M. Richardson and P. Domingos, "Markov Logic Networks," *Machine Learning J.*, vol. 62, nos. 1/2, pp. 107-136, 2006.
- [57] S. Ramachandran and R.J. Mooney, "Theory Refinement of Bayesian Networks with Hidden Variables," *Proc. 14th Int'l Conf. Machine Learning*, pp. 454-462, July 1998.
- [58] A. Arnold, R. Nallapati, and W.W. Cohen, "A Comparative Study of Methods for Transductive Transfer Learning," *Proc. Seventh IEEE Int'l Conf. Data Mining Workshops*, pp. 77-82, 2007.
- [59] T. Joachims, "Transductive Inference for Text Classification Using Support Vector Machines," *Proc. 16th Int'l Conf. Machine Learning*, pp. 200-209, 1999.
- [60] V.N. Vapnik, *Statistical Learning Theory*. Wiley Interscience, Sept. 1998.
- [61] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of Representations for Domain Adaptation," *Proc. 20th Ann. Conf. Neural Information Processing Systems*, pp. 137-144, 2007.
- [62] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning Bounds for Domain Adaptation," *Proc. 21st Ann. Conf. Neural Information Processing Systems*, pp. 129-136, 2008.
- [63] D. Xing, W. Dai, G.-R. Xue, and Y. Yu, "Bridged Refinement for Transfer Learning," *Proc. 11th European Conf. Principles and Practice of Knowledge Discovery in Databases*, pp. 324-335, Sept. 2007.
- [64] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Spectral Domain-Transfer Learning," *Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining*, pp. 488-496, Aug. 2008.
- [65] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-Bridged PLSA for Cross-Domain Text Classification," *Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 627-634, July 2008.
- [66] S.J. Pan, J.T. Kwok, and Q. Yang, "Transfer Learning via Dimensionality Reduction," *Proc. 23rd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*, pp. 677-682, July 2008.
- [67] S.J. Pan, I.W. Tsang, J.T. Kwok, and Q. Yang, "Domain Adaptation via Transfer Component Analysis," *Proc. 21st Int'l Joint Conf. Artificial Intelligence*, 2009.
- [68] M.M.H. Mahmud and S.R. Ray, "Transfer Learning Using Kolmogorov Complexity: Basic Theory and Empirical Evaluations," *Proc. 20th Ann. Conf. Neural Information Processing Systems*, pp. 985-992, 2008.
- [69] E. Eaton, M. desJardins, and T. Lane, "Modeling Transfer Relationships between Learning Tasks for Improved Inductive Transfer," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML/PKDD '08)*, pp. 317-332, Sept. 2008.
- [70] M.T. Rosenstein, Z. Marx, and L.P. Kaelbling, "To Transfer or Not to Transfer," *Proc. Conf. Neural Information Processing Systems (NIPS '05) Workshop Inductive Transfer: 10 Years Later*, Dec. 2005.
- [71] S. Ben-David and R. Schuller, "Exploiting Task Relatedness for Multiple Task Learning," *Proc. 16th Ann. Conf. Learning Theory*, pp. 825-830, 2003.
- [72] B. Bakker and T. Heskes, "Task Clustering and Gating for Bayesian Multitask Learning," *J. Machine Learning Research*, vol. 4, pp. 83-99, 2003.
- [73] A. Argyriou, A. Maurer, and M. Pontil, "An Algorithm for Transfer Learning in a Heterogeneous Environment," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML/PKDD '08)*, pp. 71-85, Sept. 2008.
- [74] R. Raina, A.Y. Ng, and D. Koller, "Constructing Informative Priors Using Transfer Learning," *Proc. 23rd Int'l Conf. Machine Learning*, pp. 713-720, June 2006.
- [75] J. Yin, Q. Yang, and L.M. Ni, "Adaptive Temporal Radio Maps for Indoor Location Estimation," *Proc. Third IEEE Int'l Conf. Pervasive Computing and Comm.*, Mar. 2005.
- [76] S.J. Pan, J.T. Kwok, Q. Yang, and J.J. Pan, "Adaptive Localization in a Dynamic WiFi Environment through Multi-View Learning," *Proc. 22nd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*, pp. 1108-1113, July 2007.
- [77] V.W. Zheng, Q. Yang, W. Xiang, and D. Shen, "Transferring Localization Models over Time," *Proc. 23rd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*, pp. 1421-1426, July 2008.

- [78] S.J. Pan, D. Shen, Q. Yang, and J.T. Kwok, "Transferring Localization Models across Space," *Proc. 23rd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*, pp. 1383-1388, July 2008.
- [79] V.W. Zheng, S.J. Pan, Q. Yang, and J.J. Pan, "Transferring Multi-Device Localization Models Using Latent Multi-Task Learning," *Proc. 23rd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*, pp. 1427-1432, July 2008.
- [80] H. Zhuo, Q. Yang, D.H. Hu, and L. Li, "Transferring Knowledge from Another Domain for Learning Action Models," *Proc. 10th Pacific Rim Int'l Conf. Artificial Intelligence*, Dec. 2008.
- [81] V.C. Raykar, B. Krishnapuram, J. Bi, M. Dundar, and R.B. Rao, "Bayesian Multiple Instance Learning: Automatic Feature Selection and Inductive Transfer," *Proc. 25th Int'l Conf. Machine Learning*, pp. 808-815, July 2008.
- [82] X. Ling, G.-R. Xue, W. Dai, Y. Jiang, Q. Yang, and Y. Yu, "Can Chinese Web Pages be Classified with English Data Source?" *Proc. 17th Int'l Conf. World Wide Web*, pp. 969-978, Apr. 2008.
- [83] Q. Yang, S.J. Pan, and V.W. Zheng, "Estimating Location Using Wi-Fi," *IEEE Intelligent Systems*, vol. 23, no. 1, pp. 8-13, Jan./Feb. 2008.
- [84] X. Shi, W. Fan, and J. Ren, "Actively Transfer Domain Knowledge," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML/PKDD '08)*, pp. 342-357, Sept. 2008.
- [85] G. Kuhlmann and P. Stone, "Graph-Based Domain Mapping for Transfer Learning in General Games," *Proc. 18th European Conf. Machine Learning*, pp. 188-200, Sept. 2007.
- [86] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, "Translated Learning," *Proc. 21st Ann. Conf. Neural Information Processing Systems*, 2008.
- [87] B. Li, Q. Yang, and X. Xue, "Transfer Learning for Collaborative Filtering via a Rating-Matrix Generative Model," *Proc. 26th Int'l Conf. Machine Learning*, June 2009.
- [88] B. Li, Q. Yang, and X. Xue, "Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction," *Proc. 21st Int'l Joint Conf. Artificial Intelligence*, July 2009.



AAAI. More details about his research and background can be found at <http://www.cse.ust.hk/~sinnopan>.



**Qiang Yang** received the bachelor's degree from Peking University in astrophysics and the PhD degree in computer science from the University of Maryland, College Park. He is a faculty member in Hong Kong University of Science and Technology's Department of Computer Science and Engineering. His research interests are data mining and machine learning, AI planning, and sensor-based activity recognition. He is a fellow of the IEEE, a member of the AAAI and the ACM, a former associate editor for the *IEEE Transactions on Knowledge and Data Engineering*, and a current associate editor for the *IEEE Intelligent Systems*. More details about his research and background can be found at <http://www.cse.ust.hk/~qyang>.

► **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**