# 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, multi-task learning and sample selection bias, as well as co-variate 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 re-collect the needed training data and rebuild the models. It would be nice to reduce the need and effort to re-collect 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 Web pages that are associated with category information obtained through previous manual-labeling efforts. For a classification task on a newly created Web site 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 Web-page classifiers learned on the university Web site to the new Web site. 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 re-calibration 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 article, 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.,

*Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clearwater Bay, Kowloon, Hong Kong*
*Emails: {sinnopan, qyang}@cse.ust.hk*

[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 Figure 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 article with a discussion of future works in Section 8.

## 2 OVERVIEW

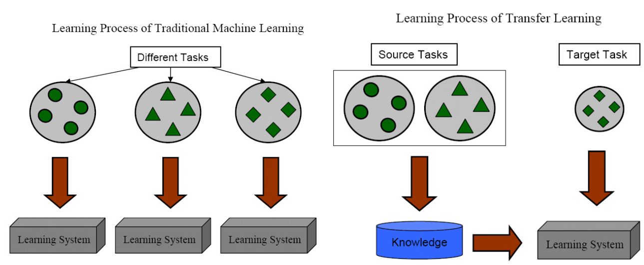
### 2.1 A Brief History of Transfer Learning

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]. Semi-supervised 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 semi-supervised learning for imperfect datasets 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" [1], 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 transfer, multi-task learning, knowledge consolidation, context-sensitive learning, knowledge-based inductive bias, meta learning, and incremental/cumulative learning [20]. Among these, a closely related learning technique to transfer learning is the multi-task learning framework [21], which tries to learn multiple tasks simultaneously even when they are different. A typical approach for multi-task 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) [2] 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 multi-task 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.

Figure 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.




(a) Traditional Machine Learning (b) Transfer Learning

Fig. 1. Different Learning Processes between Traditional Machine Learning and Transfer Learning

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) [3]. Before we give different categorizations of transfer learning, we first describe the notations used in this article.

### 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 $X = \{x_1, \ldots, x_n\} \in \mathcal{X}$. For example, if our learning

1. http://socrates.acadiau.ca/courses/comp/dsilver/NIPS95_LTL/transfer.workshop.1995.html

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

task 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 $D_S = \{(x_{S_1}, y_{S_1}), \ldots, (x_{S_{n_S}}, y_{S_{n_S}})\}$, 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 $D_T = \{(x_{T_1}, y_{T_1}), \ldots, (x_{T_{n_T}}, y_{T_{n_T}})\}$, 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 ten 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; (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 categorize transfer learning under three sub-settings, *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:

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 | *Inductive Transfer Learning* / | the same | different but related |
| | *Unsupervised Transfer Learning* | different but related | different but related |
| | *Transductive Transfer Learning* | different but related | the same |

(1.1) A lot of labeled data in the source domain are available. In this case, the *inductive transfer learning* setting is similar to the multi-task 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 multi-task learning tries to learn the target and source task simultaneously.

(1.2) 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.

(2.1) The feature spaces between the source and target domains are different, $\mathcal{X}_S \neq \mathcal{X}_T$.

(2.2) 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, $P(X_S) \neq P(X_T)$.

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 co-variate 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 Figure 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 *re-weighting*. Instance re-weighting 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 hyper-parameters 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 behind this context is that some relationship among the data in the source and target domains are 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

TABLE 2
Different Settings of Transfer Learning

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

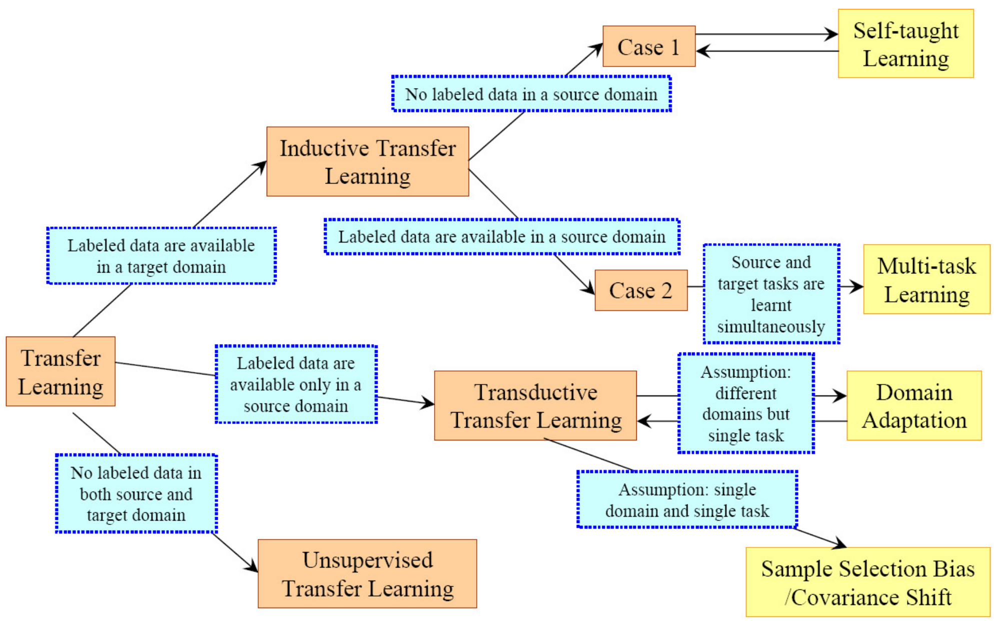



Fig. 2. An Overview of Different Settings of Transfer

TABLE 3
Different Approaches to Transfer Learning

| Transfer Learning Approaches | Brief Description |
|---|---|
| *Instance-transfer* | 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]. |
| *Feature-representation-transfer* | 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]. |
| *Parameter-transfer* | Discover shared parameters or priors between the source domain and target domain models, which can benefit for transfer learning [45], [46], [47], [48], [49]. |
| *Relational-knowledge-transfer* | 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 |
|---|---|---|---|
| *Instance-transfer* | √ | √ | |
| *Feature-representation-transfer* | √ | √ | √ |
| *Parameter-transfer* | √ | | |
| *Relational-knowledge-transfer* | √ | | |

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; (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 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 re-weight 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* in 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 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 multi-task 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 multi-task 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 multi-task learning. In the *inductive transfer learning* setting, the common features can be learned by solving an optimization problem, given as follows.

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

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 multi-task structure learning.

Lee *et al.* [42] proposed a convex optimization algorithm for simultaneously learning meta-priors and feature weights from an ensemble of related prediction tasks. The meta-priors can be transferred among different tasks. Jebara [43] proposed to select features for multi-task learning with SVMs. Ruckert *et al.* [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 $b = \{b_1, b_2, \ldots, b_s\}$ are learned on the source domain data by solving the optimization problem (2) as shown as follows,

$$\begin{aligned} \min_{a,b} \sum_i \quad & \|x_{S_i} - \textstyle\sum_j a_{S_i}^j b_j\|_2^2 + \beta\|a_{S_i}\|_1 \\ s.t. \quad & \|b_j\|_2 \le 1, \forall j \in 1,\ldots,s \end{aligned} \quad (2)$$

In this equation, $a^j_{S_i}$ 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 $b$.

$$a^*_{T_i} = \arg\min_{a_{T_i}} \|x_{T_i} - \sum_j a^j_{T_i} b_j\|_2^2 + \beta\|a_{T_i}\|_1 \qquad (3)$$

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 hyper-parameters. Most approaches described in this section, including a regularization framework and a hierarchical Bayesian framework, are designed to work under multi-task learning. However, they can be easily modified for transfer learning. As mentioned above, multi-task 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 multi-task 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 multi-task 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 multi-task learning in the context of GP. The authors proposed to use a free-form covariance matrix over tasks to model inter-task 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 multi-task 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 multi-task 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*,

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

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

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

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 loglikelihood 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 [4], Mihalkova *et al.* [51] extended *TAMAR* to the single-entity-centered setting of transfer learning, where only one entity in a target domain is available. Davis *et al.* [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 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: (a) The feature spaces between the source and target domains are different, $\mathcal{X}_S \neq \mathcal{X}_T$, and (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, $P(X_S) \neq P(X_T)$. This is similar to the requirements in domain adaptation and sample selection bias. Most approaches described in the following sections are related to case (b) 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:

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

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

4. http://www.cs.utexas.edu/∼mtaylor/AAAI08TL/

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.

$$\begin{aligned} \min_{\beta} \quad & \tfrac{1}{2}\beta^T K \beta - \kappa^T \beta \\ s.t. \quad & \beta_i \in [0, B] \;\; and \;\; |\textstyle\sum_{i=1}^{n_S} \beta_i - n_S| \leq n_S \epsilon \end{aligned} \tag{6}$$

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 $x_i \in X_S \bigcup X_T$, 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. KLIEP can be integrated with cross-validation to perform model selection automatically in two steps: (1) estimating the weights of the source domain data; (2) training models on the re-weighted data. Bickel *et al.* [33] combined the two steps in a unified framework by deriving a kernel-logistic regression classifier. Besides sample re-weighting techniques, Dai *et al.* [28] extended a traditional Naive Bayesian classifier for the transductive transfer learning problems. For more information on importance sampling and re-weighting methods for co-variate 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 [5].

### 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 revelent features that may reduce the difference between the domains. The first step of SCL is to define a set of *pivot* features [6] (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,

$$f_l(x) = sgn(w_l^T \cdot x),\ l = 1, \ldots, m$$

SCL can learn a matrix $W = [w_1 w_2 \ldots w_m]$ of parameters. In the third step, singular value decomposition (SVD) is applied to matrix $W = [w_1 w_2 \ldots w_m]$. Let $W = UDV^T$, then $\theta = U^T_{[1:h,:]}$ ($h$ 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 and Schuller [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 co-clustering 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 towards 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

5. Tutorial slides can be found at http://www.cs.columbia.edu/∼fan/PPT/ICDM08SampleBias.ppt

6. The *pivot* features are domain specific and depend on prior knowledge

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 (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 $f_T(\cdot)$ [7] 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.

$$\begin{aligned} & J(\tilde{X_T}, \tilde{X_S}, \tilde{Z}) \\ = \; & I(X_T, Z) - I(\tilde{X_T}, \tilde{Z}) + \lambda \Big[ I(X_S, Z) - I(\tilde{X_S}, \tilde{Z}) \Big] \end{aligned} \tag{7}$$

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):

$$\arg\min_{\tilde{X_T}, \tilde{X_S}, \tilde{Z}} J(\tilde{X_T}, \tilde{X_S}, \tilde{Z}) \tag{8}$$

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

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

Similarly, [27] proposed a *transferred discriminative analysis* (TDA) algorithm to solve the *transfer dimensionality reduction* problem. TDA first applies clustering methods to generate pseudo-class 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], Hassan 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 others. 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 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 Web pages 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 [8], 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 emails from a group of users with corresponding labels: *spam* or *not spam*, and train a classifier based on these data. For a new email 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 re-labeling 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**, **Email**, **WiFi** and **Sen**, respectively.

- **Text** Three data sets, 20 Newsgroups, SRAA and Reuters-21578 [9], 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 sub-categories 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.
- **Email** This data set is provided by the 2006 ECML/PKDD discovery challenge.
- **WiFi** This data set is provided by the ICDM-2007 Contest [10]. The data were collected inside a building for localization around $145.5 \times 37.5\ m^2$ in two different time periods.
- **Sen** This data set was first used in [8][11]. This data set contains product reviews downloaded from Amazon.com from $4$ 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 to 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 [12] as one of the evaluation data sets. Due to the differences in the preprocessing steps of the algorithms by different researchers, it is hard to compare the proposed methods directly. Thus, we denote them by 20-Newsgroups$_1$, 20-Newsgroups$_2$ and 20-Newsgroups$_3$, respectively, and show the comparison results between the proposed transfer learning methods and non-transfer learning methods in the table.

On the 20 Newsgroups$_1$ data, Dai *et al.* [6] showed the comparison experiments between standard Support Vector Machine (SVM) and the proposed TrAdaBoost algorithm. On 20 Newsgroups$_2$, 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$_3$, 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 Kernel Mean Matching (KMM) and the proposed algorithm, Transfer Component Analysis (TCA).

8. http://www.ecmlpkdd2006.org/challenge.html
9. 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/

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 non-transfer learning methods.

**Toolboxes for Transfer Learning:** Researchers at UC Berkeley provided a MATLAB toolkit for transfer learning [13]. The toolkit contains algorithms and benchmark data sets for transfer learning. In addition, it provides a standard 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, [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 co-clustering 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 article, 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.

### Acknowledgment

We 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,” *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[2] Q. Yang and X. Wu, “10 challenging problems in data mining research,” *International Journal of 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 revisit,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 6–20, 2006.

[4] H. Al-Mubaid and S. A. Umair, “A new text categorization technique using distributional clustering and learning logic,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 9, pp. 1156–1165, 2006.

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

TABLE 5
Comparison between transfer learning and non-transfer learning methods

| Data Set (reference) | Source v.s. Target | Baselines | | TL Methods | |
|---|---|---|---|---|---|
| 20 Newsgroups$_1$ ([6]) | | **SVM** | | **TrAdaBoost** | |
| ACC (unit: %) | 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$_2$ ([84]) | | **SVM** | | **TrAdaBoost** | **AcTraK** |
| ACC (unit: %) | 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$_3$ ([49]) | | **SVM** | **LR** | **pLWE** | **LWE** |
| ACC (unit: %) | 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]) | | **SGD** | | **SCL** | **SCL-MI** |
| ACC (unit: %) | 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]) | | **RLSR** | **PCA** | **KMM** | **TCA** |
| AED (unit: meter) | Time **A** v.s. Time **B** | 6.52 | 3.16 | 5.51 | 2.37 |


[5] K. Sarinnapakorn and M. Kubat, “Combining subclassifiers in text categorization: A dst-based solution and a case study,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1638–1651, 2007.

[6] W. Dai, Q. Yang, G. Xue, and Y. Yu, “Boosting for transfer learning,” in *Proceedings of the 24th International Conference on Machine Learning*, Corvalis, Oregon, USA, June 2007, pp. 193–200.

[7] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu, “Transfer learning for wifi-based indoor localization,” in *Proceedings of the Workshop on Transfer Learning for Complex Task of the 23rd AAAI Conference on Artificial Intelligence*, Chicago, Illinois, USA, July 2008.

[8] J. Blitzer, M. Dredze, and F. Pereira, “Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification,” in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, 2007, pp. 432–439.

[9] J. Ramon, K. Driessens, and T. Croonenborghs, “Transfer learning in reinforcement learning problems through partial policy recycling,” in *ECML ’07: Proceedings of the 18th European conference on Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 699–707.

[10] M. E. Taylor and P. Stone, “Cross-domain transfer for reinforcement learning,” in *ICML ’07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM, 2007, pp. 879–886.

[11] X. Yin, J. Han, J. Yang, and P. S. Yu, “Efficient classification across multiple database relations: A crossmine approach,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 6, pp. 770–783, 2006.

[12] L. I. Kuncheva and J. J. Rodrłguez, “Classifier ensembles with a random linear oracle,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 4, pp. 500–508, 2007.

[13] E. Baralis, S. Chiusano, and P. Garza, “A lazy approach to associative classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 2, pp. 156–171, 2008.

[14] X. Zhu, “Semi-supervised learning literature survey,” University of Wisconsin–Madison, Tech. Rep. 1530, 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, no. 2-3, pp. 103–134, 2000.

[16] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, 1998, pp. 92–100.

[17] T. Joachims, “Transductive inference for text classification using support vector machines,” in *Proceedings of Sixteenth International Conference on Machine Learning*, 1999, pp. 825–830.

[18] X. Zhu and X. Wu, “Class noise handling for effective cost-sensitive learning by cost-guided iterative classification filtering,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1435–1440, 2006.

[19] Q. Yang, C. Ling, X. Chai, and R. Pan, “Test-cost sensitive classification on data with missing values,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 5, pp. 626–638, 2006.

[20] S. Thrun and L. Pratt, Eds., *Learning to learn*. Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[21] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28(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,” in *Proceedings of the 24th International Conference on Machine Learning*, Corvalis, Oregon, USA, June 2007, pp. 759–766.

[23] H. DauméIII and D. Marcu, “Domain adaptation for statistical classifiers,” *Journal of Artificial Intelligence Research*, vol. 26, pp. 101–126, 2006.

[24] B. Zadrozny, “Learning and evaluating classifiers under sample selection bias,” in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Alberta, Canada, July 2004.

[25] H. Shimodaira, “Improving predictive inference under covariate shift by

weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, pp. 227–244, 2000.

[26] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Self-taught clustering," in *Proceedings of the 25th International Conference of Machine Learning*. ACM, July 2008, pp. 200–207.

[27] Z. Wang, Y. Song, and C. Zhang, "Transferred dimensionality reduction," in *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008*. Antwerp, Belgium: Springer, September 2008, pp. 550–565.

[28] W. Dai, G. Xue, Q. Yang, and Y. Yu, "Transferring naive bayes classifiers for text classification," in *Proceedings of the 22rd AAAI Conference on Artificial Intelligence*, Vancouver, British Columbia, Canada, July 2007, pp. 540–545.

[29] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset Shift in Machine Learning*. The MIT Press, 2009.

[30] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in nlp," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 264–271.

[31] X. Liao, Y. Xue, and L. Carin, "Logistic regression with an auxiliary data source," in *Proceedings of the 21st International Conference on Machine Learning*, Bonn, Germany, August 2005, pp. 505–512.

[32] J. Huang, A. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, 2007.

[33] S. Bickel, M. Brückner, and T. Scheffer, "Discriminative learning for differing training and test distributions," in *Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM, 2007, pp. 81–88.

[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," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada, December 2008.

[35] W. Fan, I. Davidson, B. Zadrozny, and P. S. Yu, "An improved categorization of classifier's sensitivity on sample selection bias," in *Proceedings of the 5th IEEE International Conference on Data Mining*, 2005.

[36] W. Dai, G. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, California, USA, August 2007.

[37] R. K. Ando and T. Zhang, "A high-performance semi-supervised learning method for text chunking," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Morristown, NJ, USA: Association for Computational Linguistics, 2005, pp. 1–9.

[38] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the Conference on Empirical Methods in Natural Language*, Sydney, Australia, July 2006, pp. 120–128.

[39] H. Daumé III, "Frustratingly easy domain adaptation," in *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, June 2007, pp. 256–263.

[40] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Proceedings of the 19th Annual Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada, December 2007, pp. 41–48.

[41] A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying, "A spectral regularization framework for multi-task structure learning," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2008, pp. 25–32.

[42] S.-I. Lee, V. Chatalbashev, D. Vickrey, and D. Koller, "Learning a meta-level prior for feature relevance from multiple related tasks," in *Proceedings of the 24th International Conference on Machine Learning*. Corvalis, Oregon, USA: ACM, July 2007, pp. 489–496.

[43] T. Jebara, "Multi-task feature and kernel selection for svms," in *Proceedings of the 21st International Conference on Machine Learning*. Banff, Alberta, Canada: ACM, July 2004.

[44] C. Wang and S. Mahadevan, "Manifold alignment using procrustes analysis," in *Proceedings of the 25th International Conference on Machine learning*. Helsinki, Finland: ACM, July 2008, pp. 1120–1127.

[45] N. D. Lawrence and J. C. Platt, "Learning to learn with the informative vector machine," in *Proceedings of the 21st International Conference on Machine Learning*. Banff, Alberta, Canada: ACM, July 2004.

[46] E. Bonilla, K. M. Chai, and C. Williams, "Multi-task gaussian process prediction," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2008, pp. 153–160.

[47] A. Schwaighofer, V. Tresp, and K. Yu, "Learning gaussian process kernels via hierarchical bayes," in *Proceedings of the 17th Annual Conference on Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2005, pp. 1209–1216.

[48] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Seattle, Washington, USA: ACM, August 2004, pp. 109–117.

[49] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, Nevada: ACM, August 2008, pp. 283–291.

[50] L. Mihalkova, T. Huynh, and R. J. Mooney, "Mapping and revising markov logic networks for transfer learning," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, Vancouver, British Columbia, Canada, July 2007, pp. 608–614.

[51] L. Mihalkova and R. J. Mooney, "Transfer learning by mapping with minimal target data," in *Proceedings of the AAAI-2008 Workshop on Transfer Learning for Complex Tasks*, Chicago, Illinois, USA, July 2008.

[52] J. Davis and P. Domingos, "Deep transfer via second-order markov logic," in *Proceedings of the AAAI-2008 Workshop on Transfer Learning for Complex Tasks*, Chicago, Illinois, USA, July 2008.

[53] P. Wu and T. G. Dietterich, "Improving svm accuracy by training on auxiliary data sources," in *Proceedings of the 21st International Conference on Machine Learning*. Banff, Alberta, Canada: ACM, July 2004.

[54] U. Rückert and S. Kramer, "Kernel-based inductive transfer," in *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008*, ser. Lecture Notes in Computer Science. Antwerp, Belgium: Springer, September 2008, pp. 220–233.

[55] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proceedings of the the 19th Annual Conference on Neural Information Processing Systemss*. Cambridge, MA: MIT Press, 2007, pp. 801–808.

[56] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning Journal*, vol. 62, no. 1-2, pp. 107–136, 2006.

[57] S. Ramachandran and R. J. Mooney, "Theory refinement of bayesian networks with hidden variables," in *Proceedings of the 14th International Conference on Machine Learning*, Madison, Wisconson, USA, July 1998, pp. 454–462.

[58] A. Arnold, R. Nallapati, and W. W. Cohen, "A comparative study of methods for transductive transfer learning," in *Proceedings of the 7th IEEE International Conference on Data Mining Workshops*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 77–82.

[59] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proceedings of the Sixteenth International Conference on Machine Learning*, San Francisco, CA, USA, 1999, pp. 200–209.

[60] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley-Interscience, September 1998.

[61] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2007, pp. 137–144.

[62] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Proceedings of the 21st Annual Conference on Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2008, pp. 129–136.

[63] D. Xing, W. Dai, G.-R. Xue, and Y. Yu, "Bridged refinement for transfer learning," in *11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science. Warsaw, Poland: Springer, September 2007, pp. 324–335.

[64] X. Ling, W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Spectral domain-transfer learning," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, Nevada: ACM, August 2008, pp. 488–496.

[65] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-bridged plsa for cross-domain text classification," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Singapore: ACM, July 2008, pp. 627–634.

[66] S. J. Pan, J. T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, Chicago, Illinois, USA, July 2008, pp. 677–682.

[67] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, Pasadena, California, 2009.

[68] M. M. H. Mahmud and S. R. Ray, "Transfer learning using kolmogorov complexity: Basic theory and empirical evaluations," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2008, pp. 985–992.

[69] E. Eaton, M. desJardins, and T. Lane, "Modeling transfer relationships between learning tasks for improved inductive transfer," in *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008*, ser. Lecture Notes in Computer Science. Antwerp, Belgium: Springer, September 2008, pp. 317–332.

[70] M. T. Rosenstein, Z. Marx, and L. P. Kaelbling, "To transfer or not to transfer," in *a NIPS-05 Workshop on Inductive Transfer: 10 Years Later*, December 2005.

[71] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *Proceedings of the Sixteenth Annual Conference on Learning Theory*. San Francisco: Morgan Kaufmann, 2003, pp. 825–830.

[72] B. Bakker and T. Heskes, "Task clustering and gating for bayesian multitask learning," *Journal of Machine Learning Reserch*, vol. 4, pp. 83–99, 2003.

[73] A. Argyriou, A. Maurer, and M. Pontil, "An algorithm for transfer learning in a heterogeneous environment," in *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008*, ser. Lecture Notes in Computer Science. Antwerp, Belgium: Springer, September 2008, pp. 71–85.

[74] R. Raina, A. Y. Ng, and D. Koller, "Constructing informative priors using transfer learning." in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, Pennsylvania, USA, June 2006, pp. 713–720.

[75] J. Yin, Q. Yang, and L. M. Ni, "Adaptive temporal radio maps for indoor location estimation," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, Kauai Island, Hawaii, USA, March 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," in *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, Vancouver, British Columbia, Canada, July 2007, pp. 1108–1113.

[77] V. W. Zheng, Q. Yang, W. Xiang, and D. Shen, "Transferring localization models over time," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, Chicago, Illinois, USA, July 2008, pp. 1421–1426.

[78] S. J. Pan, D. Shen, Q. Yang, and J. T. Kwok, "Transferring localization models across space," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, Chicago, Illinois, USA, July 2008, pp. 1383–1388.

[79] V. W. Zheng, S. J. Pan, Q. Yang, and J. J. Pan, "Transferring multi-device localization models using latent multi-task learning," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, Chicago, Illinois, USA, July 2008, pp. 1427–1432.

[80] H. Zhuo, Q. Yang, D. H. Hu, and L. Li, "Transferring knowledge from another domain for learning action models," in *Proceedings of 10th Pacific Rim International Conference on Artificial Intelligence*, December 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," in *Proceedings of the 25th International Conference on Machine learning*. Helsinki, Finland: ACM, July 2008, pp. 808–815.

[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?" in *Proceedings of the 17th International Conference on World Wide Web*. Beijing, China: ACM, April 2008, pp. 969–978.

[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, 2008.

[84] X. Shi, W. Fan, and J. Ren, "Actively transfer domain knowledge," in *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML/PKDD 2008*, ser. Lecture Notes in Computer Science. Antwerp, Belgium: Springer, September 2008, pp. 342–357.

[85] G. Kuhlmann and P. Stone, "Graph-based domain mapping for transfer learning in general games," in *18th European Conference on Machine Learning*, ser. Lecture Notes in Computer Science. Warsaw, Poland: Springer, September 2007, pp. 188–200.

[86] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, "Translated learning," in *Proceedings of 21st Annual Conference on Neural Information Processing Systems*, 2008.

[87] B. Li, Q. Yang, and X. Xue, "Transfer learning for collaborative filtering via a rating-matrix generative model," in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Quebec, Canada, June 2009.

[88] B. Li, Q. Yang, and X. Xue, "Can movies and books collaborate? - cross-domain collaborative filtering for sparsity reduction," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, Pasadena, California, USA, July 2009.

**Sinno Jialin Pan** is a PhD candidate in the Department of Computer Science and Engineering, the Hong Kong University of Science and Technology. He received the MS and BS degrees from Applied Mathematics Department, Sun Yat-sen University, China, in 2003 and 2005, respectively. His research interests include transfer learning, semi-supervised learning, and their applications in pervasive computing and Web mining. He is a member of AAAI. Contact him at the Department of Computer Science and Engineering, Hong Kong Univ. of Science and Technology, Clearwater Bay, Kowloon, Hong Kong; sinnopan@cse.ust.hk; http://www.cse.ust.hk/∼sinnopan.

**Qiang Yang** is a faculty member in the 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 received his PhD degree in Computer Science from the University of Maryland, College Park, and Bachelor's degree from Peking University in Astrophysics. He is a fellow of IEEE, member of AAAI and ACM, a former associate editor for the IEEE TKDE, and a current associate editor for IEEE Intelligent Systems. Contact him at the Department of Computer Science and Engineering, Hong Kong Univ. of Science and Technology, Clearwater Bay, Kowloon, Hong Kong; qyang@cse.ust.hk; http://www.cse.ust.hk/∼qyang.