A Comprehensive Survey on Transfer Learning

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, *Senior Member, IEEE,*
Hui Xiong, *Fellow, IEEE,* and Qing He

Abstract—Transfer learning aims at improving the performance of target learners on target domains by transferring the knowledge contained in different but related source domains. In this way, the dependence on a large number of target domain data can be reduced for constructing target learners. Due to the wide application prospects, transfer learning has become a popular and promising area in machine learning. Although there are already some valuable and impressive surveys on transfer learning, these surveys introduce approaches in a relatively isolated way and lack the recent advances in transfer learning. Due to the rapid expansion of the transfer learning area, it is both necessary and challenging to comprehensively review the relevant studies. This survey attempts to connect and systematize the existing transfer learning researches, as well as to summarize and interpret the mechanisms and the strategies of transfer learning in a comprehensive way, which may help readers have a better understanding of the current research status and ideas. Unlike previous surveys, this survey paper reviews more than forty representative transfer learning approaches, especially homogeneous transfer learning approaches, from the perspectives of data and model. The applications of transfer learning are also briefly introduced. In order to show the performance of different transfer learning models, over twenty representative transfer learning models are used for experiments. The models are performed on three different datasets, i.e., Amazon Reviews, Reuters-21578, and Office-31. And the experimental results demonstrate the importance of selecting appropriate transfer learning models for different applications in practice.

Index Terms—Transfer learning, machine learning, domain adaptation, interpretation.

✦

1 INTRODUCTION

A LTHOUGH traditional machine learning technology has achieved great success and has been successfully applied in many practical applications, it still has some limitations for certain real-world scenarios. The ideal scenario of machine learning is that there are abundant labeled training instances, which have the same distribution as the test data. However, collecting sufficient training data is often expensive, time-consuming, or even unrealistic in many scenarios. Semi-supervised learning can partly solve this problem by relaxing the need of mass labeled data. Typically, a semi-supervised approach only requires a limited number of labeled data, and it utilizes a large amount of unlabeled data to improve the learning accuracy. But in many cases, unlabeled instances are also difficult to collect, which usually makes the resultant traditional models unsatisfactory.

Transfer learning, which focuses on transferring the knowledge across domains, is a promising machine learning methodology for solving the above problem. The concept about transfer learning may initially come from educational psychology. According to the generalization theory of transfer, as proposed by psychologist C.H. Judd, learning to transfer is the result of the generalization of experience. It is possible to realize the transfer from one situation to another,
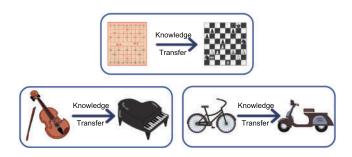

Fig. 1. Intuitive examples about transfer learning.

as long as a person generalizes his experience. According to this theory, the prerequisite of transfer is that there needs to be a connection between two learning activities. In practice, a person who has learned the violin can learn the piano faster than others, since both the violin and the piano are musical instruments and may share some common knowledge. Fig. 1 shows some intuitive examples about transfer learning. Inspired by human beings' capabilities to transfer knowledge across domains, transfer learning aims to leverage knowledge from a related domain (called source domain) to improve the learning performance or minimize the number of labeled examples required in a target domain. It is worth mentioning that the transferred knowledge does not always bring a positive impact on new tasks. If there is little in common between domains, knowledge transfer could be unsuccessful. For example, learning to ride a bicycle cannot help us learn to play the piano faster. Besides, the similarities between domains do not always facilitate learning, because sometimes the similarities may be misleading. For example, although Spanish and French have a close relationship with each other and both belong to the Romance

• *Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, and Qing He are with the Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China and the University of Chinese Academy of Sciences, Beijing 100049, China.*
• *Hengshu Zhu is with Baidu Inc., No. 10 Shangdi 10th Street, Haidian District, Beijing, China.*
• *Hui Xiong is with Rutgers, the State University of New Jersey, 1 Washington Park, Newark, New Jersey, USA.*
• *Zhiyuan Qi is with the equal contribution to the first author.*
• *Fuzhen Zhuang and Zhiyuan Qi are corresponding authors, zhuangfuzhen@ict.ac.cn and qizhyuan@gmail.com.*

group of languages, people who learn Spanish may experience difficulties in learning French, such as using the wrong vocabulary or conjugation. This occurs because previous successful experience in Spanish can interfere with learning the word formation, usage, pronunciation, conjugation, etc., in French. In the field of psychology, the phenomenon that previous experience has a negative effect on learning new tasks is called negative transfer [1]. Similarly, in the transfer learning area, if the target learner is negatively affected by the transferred knowledge, the phenomenon is also termed as negative transfer [2], [3]. Whether negative transfer will occur may depend on several factors, such as the relevance between the source and the target domains and the learners capacity of finding the transferable and beneficial part of the knowledge across domains. In [3], a formal definition and some analyses of negative transfer are given.

Roughly speaking, according to the discrepancy between domains, transfer learning can be further divided into two categories, i.e., homogeneous and heterogeneous transfer learning [4]. Homogeneous transfer learning approaches are developed and proposed for handling the situations where the domains are of the same feature space. In homogeneous transfer learning, some studies assume that domains differ only in marginal distributions. Therefore, they adapt the domains by correcting the sample selection bias [5] or covariate shift [6]. However, this assumption does not hold in many cases. For example, in sentiment classification problem, a word may have different meaning tendencies in different domains. This phenomenon is also called context feature bias [7]. To solve this problem, some studies further adapt the conditional distributions. Heterogeneous transfer learning refers to the knowledge transfer process in the situations where the domains have different feature spaces. In addition to distribution adaptation, heterogeneous transfer learning requires feature space adaptation [7], which makes it more complicated than homogeneous transfer learning.

The survey aims to give readers a comprehensive understanding about transfer learning from the perspectives of data and model. The mechanisms and the strategies of transfer learning approaches are introduced to allow readers grasp how the approaches work. And a number of the existing transfer learning researches are connected and systematized. Specifically, over forty representative transfer learning approaches are introduced. Besides, we conduct experiments to demonstrate on which dataset a transfer learning model performs well.

In this survey, we focus more on homogeneous transfer learning. Some interesting transfer learning topics are not covered in this survey, such as reinforcement transfer learning [8], lifelong transfer learning [9], and online transfer learning [10]. The rest of this survey are organized into seven sections. Section 2 clarifies the difference between transfer learning and other related machine learning techniques. Section 3 introduces the notations used in this survey and the definitions about transfer learning. Sections 4 and 5 interpret transfer learning approaches from the data and the model perspectives, respectively. Section 6 introduces some applications of transfer learning. Experiments are conducted and the results are provided in Section 7. The last section concludes this survey. The main contributions of this survey are summarized below.

- Over forty representative transfer learning approaches are introduced and summarized, which can give readers a comprehensive overview about transfer learning.
- Experiments are conducted to compare different transfer learning approaches. The performance of over twenty different approaches is displayed intuitively and then analyzed, which may be instructive for the readers to select the appropriate ones in practice.

2 RELATED WORK

Some areas related to transfer learning are introduced. The connections and difference between them and transfer learning are clarified.

Semi-Supervised Learning [11]: Semi-supervised learning is a machine learning task and method that lies between supervised learning (with completely labeled instances) and unsupervised learning (without any labeled instances). Typically, a semi-supervised method utilizes abundant unlabeled instances combined with a limited number of labeled instances to train a learner. Semi-supervised learning relaxes the dependence on labeled instances, and thus reduces the expensive labeling costs. Note that, in semi-supervised learning, both the labeled and unlabeled instances are drawn from the same distribution. In contrast, in transfer learning, the data distributions of the source and the target domains are usually different. Many transfer learning approaches absorb the technology of semi-supervised learning. The key assumptions in semi-supervised learning, i.e., smoothness, cluster, and manifold assumptions, are also made use of in transfer learning. It is worth mentioning that semi-supervised transfer learning is a controversial term. The reason is that the concept of whether the label information is available in transfer learning is ambiguous because both the source and the target domains can be involved.

Multi-View Learning [12]: Multi-view learning focuses on the machine learning problems with multi-view data. A view represents a distinct feature set. An intuitive example about multiple views is that a video object can be described from two different viewpoints, i.e., the image signal and the audio signal. Briefly, multi-view learning describes an object from multiple views, which results in abundant information. By properly considering the information from all views, the learner's performance can be improved. There are several strategies adopted in multi-view learning such as subspace learning, multi-kernel learning, and co-training [13], [14]. Multi-view techniques are also adopted in some transfer learning approaches. For example, Zhang *et al.* proposed a multi-view transfer learning framework, which imposes the consistency among multiple views [15]. Yang and Gao incorporated multi-view information across different domains for knowledge transfer [16]. The work by Feuz and Cook introduces a multi-view transfer learning approach for activity learning, which transfers activity knowledge between heterogeneous sensor platforms [17].

Multi-Task Learning [18]: The thought of multi-task learning is to jointly learn a group of related tasks. To be more specific, multi-task learning reinforces each task by taking advantage of the interconnections between task, i.e., considering both the inter-task relevance and the inter-task

difference. In this way, the generalization of each task is enhanced. The main difference between transfer learning and multi-task learning is that the former transfer the knowledge contained in the related domains, while the latter transfer the knowledge via simultaneously learning some related tasks. In other words, multi-task learning pays equal attention to each task, while transfer learning pays more attention to the target task than to the source task. There are some commons and associations between transfer learning and multi-task learning. Both of them aim to improve the performance of learners via knowledge transfer. Besides, they adopt some similar strategies for constructing models, such as feature transformation and parameter sharing. Note that some existing studies utilize both the transfer learning and the multi-task learning technologies. For example, the work by Zhang *et al.* employs multi-task and transfer learning techniques for biological image analysis [19]. The work by Liu *et al.* proposes a framework for human action recognition based on multi-task learning and multi-source transfer learning [20].

3 OVERVIEW

In this section, the notations used in this survey are listed for convenience. Besides, some definitions and categorizations about transfer learning are introduced, and some related surveys are also provided.

3.1 Notation

For convenience, a list of symbols and their definitions are shown in Table 1. Besides, we use $||\cdot||$ to represent the norm and superscript ^{T} to denote the transpose of a vector/matrix.

3.2 Definition

In this subsection, some definitions about transfer learning are given. Before giving the definition of transfer learning, let us review the definitions of a domain and a task.

Definition 1. (Domain) *A domain $\mathcal{D}$ is composed of two parts, i.e., a feature space $\mathcal{X}$ and a marginal distribution $P(X)$. In other words, $\mathcal{D} = \{\mathcal{X}, P(X)\}$. And the symbol X denotes an instance set, which is defined as $X = \{\mathbf{x}|\mathbf{x}_i \in \mathcal{X},\ i = 1, \cdots, n\}$.*

Definition 2. (Task) *A task $\mathcal{T}$ consists of a label space $\mathcal{Y}$ and a decision function f, i.e., $\mathcal{T} = \{\mathcal{Y}, f\}$. The decision function f is an implicit one, which is expected to be learned from the sample data.*

Some machine learning models actually output the predicted conditional distributions of instances. In this case, $f(\mathbf{x}_j) = \{P(y_k|\mathbf{x}_j)|y_k \in \mathcal{Y}, k = 1, \cdots, |\mathcal{Y}|\}$.

In practice, a domain is often observed by a number of instances with/without the label information. For example, a source domain $\mathcal{D}_S$ corresponding to a source task $\mathcal{T}_S$ is usually observed via the instance-label pairs, i.e., $D_S = \{(\mathbf{x}, y)|\mathbf{x}_i \in \mathcal{X}^S, y_i \in \mathcal{Y}^S, i = 1, \cdots, n^S\}$; an observation of the target domain usually consists of a number of unlabeled instances and/or limited number of labeled instances.

Definition 3. (Transfer Learning) *Given some/an observation(s) corresponding to $m^S \in \mathbb{N}^+$ source domain(s) and task(s)*

TABLE 1
Notations.

Symbol	Definition
n	Number of instances
m	Number of domains
$\mathcal{D}$	Domain
$\mathcal{T}$	Task
$\mathcal{X}$	Feature space
$\mathcal{Y}$	Label space
$\mathbf{x}$	Feature vector
y	Label
X	Instance set
Y	Label set corresponding to X
S	Source domain
T	Target domain
L	Labeled instances
U	Unlabeled instances
$\mathcal{H}$	Reproducing kernel Hilbert space
$\boldsymbol{\theta}$	Mapping/Coefficient vector
α	Weighting coefficient
β	Weighting coefficient
λ	Tradeoff parameter
δ	Parameter/Error
b	Bias
B	Boundary parameter
N	Iteration/Kernel number
f	Decision function
$\mathcal{L}$	Loss function
η	Scale parameter
G	Graph
Φ	Nonlinear mapping
σ	Monotonically increasing function
Ω	Structural risk
κ	Kernel function
K	Kernel matrix
H	Centering matrix
C	Covariance matrix
d	Document
w	Word
z	Class variable
$\tilde{\mathbf{z}}$	Noise
$\mathcal{D}$	Discriminator
$\mathcal{G}$	Generator
S	Function
M	Orthonormal bases
Θ	Model parameters
P	Probability
$\mathbb{E}$	Expectation
Q	Matrix variable
R	Matrix variable
W	Mapping matrix

(i.e., $\{(\mathcal{D}_{S_i}, \mathcal{T}_{S_i})|i = 1, \cdots, m^S\}$), and some/an observation(s) about $m^T \in \mathbb{N}^+$ target domain(s) and task(s) (i.e., $\{(\mathcal{D}_{T_j}, \mathcal{T}_{T_j})|j = 1, \cdots, m^T\}$), transfer learning utilizes the knowledge implied in the source domain(s) to improve the performance of the learned decision functions f^{T_j} ($j = 1, \cdots, m^T$) on the target domain(s).

The above definition, which covers the situation of multi-source transfer learning, is an extension of the one presented in the survey [2]. If m^S equals 1, the scenario is called single-source transfer learning. Otherwise, it is called multi-source transfer learning. Besides, m^T represents the number of the transfer learning tasks. A few studies focus on the setting that $m^T \geq 2$ [21]. The existing transfer learning studies pay more attention to the scenarios where $m^T = 1$ (especially where $m^S = m^T = 1$). It is worth mentioning that the observation of a domain or a task is a concept with broad
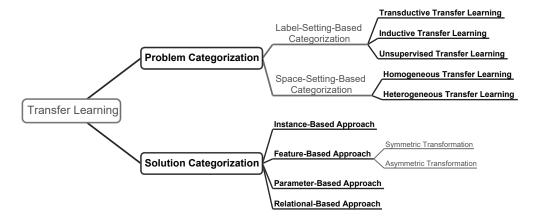
Fig. 2. Categorizations of transfer learning.

sense, which is often cemented into a labeled/unlabeled instance set or a pre-learned model. A common scenario is that we have abundant labeled instances or have a well-trained model on the source domain, and we only have limited labeled target-domain instances. In this case, the resources such as the instances and the model are actually the observations, and the goal of transfer learning is to learn a more accurate decision function on the target domain.

Another term commonly used in the transfer learning area is domain adaptation. Domain adaptation refers to the process that adapting one or more source domains to transfer knowledge and improve the performance of the target learner [4]. Transfer learning often relies on the domain adaptation process, which attempts to reduce the difference between domains.

3.3 Categorization of Transfer Learning

There are several categorization criteria of transfer learning. For example, transfer learning problems can be divided into three categories, i.e., transductive, inductive, and unsupervised transfer learning [2]. The complete definitions of these three categories are presented in [2]. These three categories can be interpreted from a label-setting aspect. Roughly speaking, transductive transfer learning refers to the situations where the label information only comes from the source domain. If the label information of the target-domain instances is available, the scenario can be categorized into inductive transfer learning. If the label information is unknown for both the source and the target domains, the situation is known as unsupervised transfer learning. Another categorization is based on the consistency between the source and the target feature spaces and label spaces. If $\mathcal{X}^S = \mathcal{X}^T$ and $\mathcal{Y}^S = \mathcal{Y}^T$, the scenario is termed as homogeneous transfer learning. Otherwise, if $\mathcal{X}^S \neq \mathcal{X}^T$ or/and $\mathcal{Y}^S \neq \mathcal{Y}^T$, the scenario is referred to as heterogeneous transfer learning.

According to the survey [2], the transfer learning approaches can be categorized into four groups: instance-based, feature-based, parameter-based, and relational-based approaches. Instance-based transfer learning approaches are mainly based on the instance weighting strategy. Feature-based approaches transform the original features to create a new feature representation; they can be further divided into

two subcategories, i.e., asymmetric and symmetric feature-based transfer learning. Asymmetric approaches transform the source features to match the target ones. In contrast, symmetric approaches attempt to find a common latent feature space and then transform both the source and the target features into a new feature representation. The parameter-based transfer learning approaches transfer the knowledge at the model/parameter level. Relational-based transfer learning approaches mainly focus on the problems in relational domains. Such approaches transfer the logical relationship or rules learned in the source domain to the target domain. For better understanding, Fig. 2 shows the above-mentioned categorizations of transfer learning.

Some surveys are provided for the readers who want to have a more complete understanding of this field. The survey by Pan and Yang [2], which is a pioneering work, categorizes transfer learning and reviews the research progress before 2010. The survey by Weiss *et al.* introduces and summarizes a number of homogeneous and heterogeneous transfer learning approaches [4]. Heterogeneous transfer learning is specially reviewed in the survey by Day and Khoshgoftaar [7]. Some surveys review the literatures related to specific themes such as reinforcement learning [8], computational intelligence [22], and deep learning [23], [24]. Besides, a number of surveys focus on specific application scenarios including activity recognition [25], visual categorization [26], collaborative recommendation [27], computer vision [24], and sentiment analysis [28].

Note that the organization of this survey does not strictly follow the above-mentioned categorizations. In the next two sections, transfer learning approaches are interpreted from the data and the model perspectives. Roughly speaking, data-based interpretation covers the above-mentioned instance-based and feature-based transfer learning approaches, but from a broader perspective. Model-based interpretation covers the above-mentioned parameter-based approaches. Since there are relatively few studies concerning relational-based transfer learning and the representative approaches are well introduced in [2], [4], this survey does not focus on relational-based approaches.
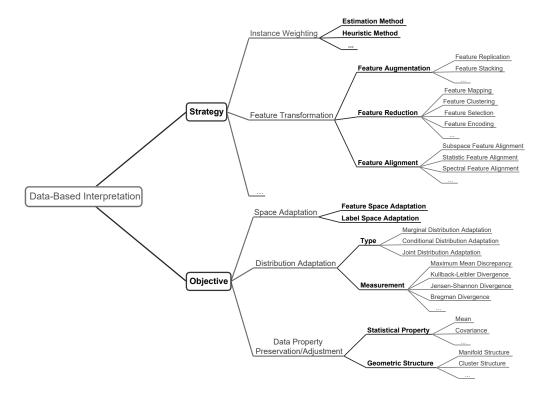
Fig. 3. Strategies and the objectives of the transfer learning approaches from the data perspective.

4 DATA-BASED INTERPRETATION

Many transfer learning approaches, especially the data-based approaches, focus on transferring the knowledge via the adjustment and the transformation of data. Fig. 3 shows the strategies and the objectives of the approaches from the data perspective. As shown in Fig. 3, space adaptation is one of the objectives. This objective is required to be satisfied mostly in heterogeneous transfer learning scenarios. In this survey, we focus more on homogeneous transfer learning, and the main objective in this scenario is to reduce the distribution difference between the source-domain and the target-domain instances. Furthermore, some advanced approaches may attempt to preserve the data properties in the adaptation process. There are generally two strategies for realizing the objectives from the data perspective, i.e., instance weighting and feature transformation. In this section, some related transfer learning approaches are introduced in proper order according to the strategies shown in Fig. 3.

4.1 Instance Weighting Strategy

Let us first consider a simple scenario in which a large number of labeled source-domain and a limited number of target-domain instances are available and domains differ only in marginal distributions (i.e., $P^S(X) \neq P^T(X)$ and $P^S(Y|X) = P^T(Y|X)$). For example, suppose we need to build a model to diagnose cancer in a specific region where the elderly are the majority. Limited target-domain instances are given, and relevant data are available from another region where young people are the majority. Directly transferring all the data from another region may be unsuccessful, since the marginal distribution difference exists, and the elderly have a higher risk of cancer than younger people. In this scenario, it is natural to consider

adapting the marginal distributions. A simple idea is to assign weights to the source-domain instances in the loss function. The weighting strategy is based on the following equation [5]:

$$\mathbb{E}_{(\mathbf{x},y)\sim P^T}\left[\mathcal{L}(\mathbf{x},y;f)\right] = \mathbb{E}_{(\mathbf{x},y)\sim P^S}\left[\frac{P^T(\mathbf{x},y)}{P^S(\mathbf{x},y)}\mathcal{L}(\mathbf{x},y;f)\right]$$
$$= \mathbb{E}_{(\mathbf{x},y)\sim P^S}\left[\frac{P^T(\mathbf{x})}{P^S(\mathbf{x})}\mathcal{L}(\mathbf{x},y;f)\right].$$

Therefore, the general objective function of a learning task can be written as [5]:

$$\min_f \frac{1}{n^S}\sum_{i=1}^{n^S}\beta_i\mathcal{L}\left(f(\mathbf{x}_i^S),y_i^S\right) + \Omega(f),$$

where β_i $(i = 1,2,\cdots,n^S)$ is the weighting parameter. The theoretical value of β_i is equal to $P^T(\mathbf{x}_i)/P^S(\mathbf{x}_i)$. However, this ratio is generally unknown and is difficult to be obtained by using the traditional methods.

Kernel Mean Matching (KMM) [5], which is proposed by Huang *et al.*, resolves the estimation problem of the above unknown ratios by matching the means between the source-domain and the target-domain instances in a Reproducing Kernel Hilbert Space (RKHS), i.e.,

$$\arg\min_{\beta_i\in[0,B]}\left\|\frac{1}{n^S}\sum_{i=1}^{n^S}\beta_i\Phi(\mathbf{x}_i^S) - \frac{1}{n^T}\sum_{j=1}^{n^T}\Phi(\mathbf{x}_j^T)\right\|_{\mathcal{H}}^2$$
$$s.t.\ |\frac{1}{n^S}\sum_{i=1}^{n^S}\beta_i - 1| \leq \delta,$$

where δ is a small parameter, and B is a parameter for constraint. The above optimization problem can be converted

into a quadratic programming problem by expanding and using the kernel trick. This approach to estimating the ratios of distributions can be easily incorporated into many existing algorithms. Once the weight β_i is obtained, a learner can be trained on the weighted source-domain instances.

There are some other studies attempting to estimate the weights. For example, Sugiyama *et al.* proposed an approach termed Kullback-Leibler Importance Estimation Procedure (KLIEP) [6]. KLIEP depends on the minimization of the Kullback-Leibler (KL) divergence and incorporates a built-in model selection procedure. Based on the studies of weight estimation, some instance-based transfer learning frameworks or algorithms are proposed. For example, Sun *et al.* proposed a multi-source framework termed 2-Stage Weighting Framework for Multi-Source Domain Adaptation (2SW-MDA) with the following two stages [29].

1. *Instance Weighting*: The source-domain instances are assigned with weights to reduce the marginal distribution difference, which is similar to KMM.
2. *Domain Weighting*: Weights are assigned to each source domain for reducing the conditional distribution difference based on the smoothness assumption [30].

Then, the source-domain instances are reweighted based on the instance weights and the domain weights. These reweighted instances and the labeled target-domain instances are used to train the target classifier.

In addition to directly estimating the weighting parameters, adjusting weights iteratively is also effective. The key is to design a mechanism to decrease the weights of the instances that have negative effects on the target learner. A representative work is TrAdaBoost [31], which is a framework proposed by Dai *et al*. This framework is an extension of AdaBoost [32]. AdaBoost is an effective boosting algorithm designed for traditional machine learning tasks. In each iteration of AdaBoost, a learner is trained on the instances with updated weights, which results in a weak classifier. The weighting mechanism of instances ensures that the instances with incorrect classification are given more attention. Finally, the resultant weak classifiers are combined to form a strong classifier. TrAdaBoost extends the AdaBoost to the transfer learning scenario; a new weighting mechanism is designed to reduce the impact of the distribution difference. Specifically, in TrAdaBoost, the labeled source-domain and labeled target-domain instances are combined as a whole, i.e., a training set, to train the weak classifier. The weighting operations are different for the source-domain and the target-domain instances. In each iteration, a temporary variable $\bar{\delta}$, which measures the classification error rate on the labeled target-domain instances, is calculated. Then, the weights of the target-domain instances are updated based on $\bar{\delta}$ and the individual classification results, while the weights of the source-domain instances are updated based on a designed constant and the individual classification results. For better understanding, the formulas used in the k-th iteration ($k = 1, \cdots, N$) for updating the weights are presented repeatedly as follows [31]:

$$\beta_{k,i}^S = \beta_{k-1,i}^S (1 + \sqrt{2 \ln n^S / N})^{-|f_k(\mathbf{x}_i^S) - y_i^S|} \ (i = 1, \cdots, n^S),$$
$$\beta_{k,j}^T = \beta_{k-1,j}^T (\bar{\delta}_k / (1 - \bar{\delta}_k))^{-|f_k(\mathbf{x}_j^T) - y_j^T|} \ (j = 1, \cdots, n^T).$$

Note that each iteration forms a new weak classifier. The final classifier is constructed by combining and ensembling half the number of the newly resultant weak classifiers through voting scheme.

Some studies further extend TrAdaBoost. The work by Yao and Doretto [33] proposes a Multi-Source TrAdaBoost (MsTrAdaBoost) algorithm, which mainly has the following two steps in each iteration.

1. *Candidate Classifier Construction*: A group of candidate weak classifiers are respectively trained on the weighted instances in the pairs of each source domain and the target domain, i.e., $\mathcal{D}_{S_i} \cup \mathcal{D}_T$ ($i = 1, \cdots, m^S$).
2. *Instance Weighting*: A classifier (denoted by j and trained on $\mathcal{D}_{S_j} \cup \mathcal{D}_T$) which has the minimal classification error rate $\bar{\delta}$ on the target domain instances is selected, and then is used for updating the weights of the instances in $\mathcal{D}_{S_j}$ and $\mathcal{D}_T$.

Finally, the selected classifiers from each iteration are combined to form the final classifier. Another parameter-based algorithm, i.e., TaskTrAdaBoost, is also proposed in the work [33], which is introduced in Section 5.3.

Some approaches realize instance weighting strategy in a heuristic way. For example, Jiang and Zhai proposed a general weighting framework [34]. There are three terms in the framework's objective function, which are designed to minimize the cross-entropy loss of three types of instances. The following types of instances are used to construct the target classifier.

- *Labeled Target-domain Instance*: The classifier should minimize the cross-entropy loss on them, which is actually a standard supervised learning task.
- *Unlabeled Target-domain Instance*: These instances' true conditional distributions $P(y|\mathbf{x}_i^{T,U})$ are unknown and should be estimated. A possible solution is to train an auxiliary classifier on the labeled source-domain and target-domain instances to help estimate the conditional distributions or assign pseudo labels to these instances.
- *Labeled Source-domain Instance*: The authors define the weight of $\mathbf{x}_i^{S,L}$ as the product of two parts, i.e., α_i and β_i. The weight β_i is ideally equal to $P^T(\mathbf{x}_i)/P^S(\mathbf{x}_i)$, which can be estimated by non-parametric methods such as KMM or can be set uniformly in the worst case. The weight α_i is used to filter out the source-domain instances that differ greatly from the target domain.

A heuristic method can be used to produce the value of α_i, which contains the following three steps.

1. *Auxiliary Classifier Construction*: An auxiliary classifier trained on the labeled target-domain instances are used to classify the unlabeled source-domain instances.
2. *Instance Ranking*: The source-domain instances are ranked based on the probabilistic prediction results.
3. *Heuristic Weighting* (β_i): The weights of the top-k source-domain instances with wrong predictions are set to zero, and the weights of others are set to one.

4.2 Feature Transformation Strategy

Feature transformation strategy is often adopted in feature-based approaches. For example, consider a cross-domain text classification problem. The task is to construct a target

TABLE 2
Metrics Adopted in Transfer Learning.

Measurement	Related Algorithms
Maximum Mean Discrepancy [35]	[36] [37] [38] [39]···
Kullback-Leibler Divergence [40]	[41] [42] [43] [44]···
Jensen-Shannon Divergence [45]	[46] [47] [48] [49]···
Bregman Divergence [50]	[51] [52] [53] [54]···
Hilbert-Schmidt Independence Criterion [55]	[36] [56] [57] [58]···

classifier by using the labeled text data from a related domain. In this scenario, a feasible solution is to find the common latent features (e.g., latent topics) through feature transformation and use them as a bridge to transfer knowledge. Feature-based approaches transform each original feature into a new feature representation for knowledge transfer. The objectives of constructing a new feature representation include minimizing the marginal and the conditional distribution difference, preserving the properties or the potential structures of the data, and finding the correspondence between features. The operations of feature transformation can be divided into three types, i.e., feature augmentation, feature reduction, and feature alignment. Besides, feature reduction can be further divided into several types such as feature mapping, feature clustering, feature selection, and feature encoding. A complete feature transformation process designed in an algorithm may consist of several operations.

4.2.1 Distribution Difference Metric

One primary objective of feature transformation is to reduce the distribution difference of the source and the target domain instances. Therefore, how to measure the distribution difference or the similarity between domains effectively is an important issue.

The measurement termed Maximum Mean Discrepancy (MMD) is widely used in the field of transfer learning, which is formulated as follows [35]:

$$\text{MMD}(X^S, X^T) = \left\| \frac{1}{n^S} \sum_{i=1}^{n^S} \Phi(\mathbf{x}_i^S) - \frac{1}{n^T} \sum_{j=1}^{n^T} \Phi(\mathbf{x}_j^T) \right\|_{\mathcal{H}}^2.$$

MMD can be easily computed by using kernel trick. Briefly, MMD quantifies the distribution difference by calculating the distance of the mean values of the instances in a RKHS. Note that the above-mentioned KMM actually produces the weights of instances by minimizing the MMD distance between domains.

Table. 2 lists some commonly used metrics and the related algorithms. In addition to Table. 2, there are some other measurement criteria adopted in transfer learning, including Wasserstein distance [59], [60], central moment discrepancy [61], etc. Some studies focus on optimizing and improving the existing measurements. Take MMD as an example. Gretton *et al.* proposed a multi-kernel version of MMD, i.e., MK-MMD [62], which takes advantage of multiple kernels. Besides, Yan *et al.* proposed a weighted version of MMD [63], which attempts to address the issue of class weight bias.

4.2.2 Feature Augmentation

Feature augmentation operations are widely used in feature transformation, especially in symmetric feature-based approaches. To be more specific, there are several ways to realize feature augmentation such as feature replication and feature stacking. For better understanding, we start with a simple transfer learning approach which is established based on feature replication.

The work by Daumé proposes a simple domain adaptation method, i.e., Feature Augmentation Method (FAM) [64]. This method transforms the original features by simple feature replication. Specifically, in single-source transfer learning scenario, the feature space is augmented to three times its original size. The new feature representation consists of general features, source-specific features, and target-specific features. Note that, for the transformed source-domain instances, their target-specific features are set to zero. Similarly, for the transformed target-domain instances, their source-specific features are set to zero. The new feature representation of FAM is presented as follows:

$$\Phi_S(\mathbf{x}_i^S) = \langle \mathbf{x}_i^S, \mathbf{x}_i^S, \mathbf{0} \rangle, \ \Phi_T(\mathbf{x}_j^T) = \langle \mathbf{x}_j^T, \mathbf{0}, \mathbf{x}_j^T \rangle,$$

where Φ_S and Φ_T denote the mappings to the new feature space from the source and the target domain, respectively. The final classifier is trained on the transformed labeled instances. It is worth mentioning that this augmentation method is actually redundant. In other words, augmenting the feature space in other ways (with fewer dimensions) may be able to produce competent performance. The superiority of FAM is that its feature expansion has an elegant form, which results in some good properties such as the generalization to multi-source scenarios. An extension of FAM is proposed in [65] by Daumé *et al.*, which utilizes the unlabeled instances to further facilitate the knowledge transfer process.

However, FAM may not work well in handling heterogeneous transfer learning tasks. The reason is that directly replicating features and padding zero vectors are less effective when the source and the target domains have different feature representations. To solve this problem, Li *et al.* proposed an approach termed Heterogeneous Feature Augmentation (HFA) [66], [67]. The feature representation of HFA is presented below:

$$\Phi_S(\mathbf{x}_i^S) = \langle W^S \mathbf{x}_i^S, \mathbf{x}_i^S, \mathbf{0}^T \rangle, \ \Phi_T(\mathbf{x}_j^T) = \langle W^T \mathbf{x}_j^T, \mathbf{0}^S, \mathbf{x}_j^T \rangle,$$

where $W^S \mathbf{x}_i^S$ and $W^T \mathbf{x}_j^T$ have the same dimension; $\mathbf{0}^S$ and $\mathbf{0}^T$ denote the zero vectors with the dimensions of $\mathbf{x}^S$ and $\mathbf{x}^T$, respectively. HFA maps the original features into a common feature space, and then performs a feature stacking operation. The mapped features, original features, and zero elements are stacked in a particular order to produce a new feature representation.

4.2.3 Feature Mapping

In the field of traditional machine learning, there are many feasible mapping-based methods of extracting features such as Principal Component Analysis (PCA) [68] and Kernelized-PCA (KPCA) [69]. However, these methods mainly focus on the data variance rather than the distribution difference. In order to solve the distribution difference,

some feature extraction methods are proposed for transfer learning. Let us first consider a simple scenario where there is little difference in the conditional distributions of the domains. In this case, the following simple objective function can be used to find a mapping for feature extraction:

$$\min_{\Phi} \big(\text{DIST}(X^S, X^T; \Phi) + \lambda\Omega(\Phi)\big)/\big(\text{VAR}(X^S \cup X^T; \Phi)\big),$$

where Φ is a low-dimensional mapping function, $\text{DIST}(\cdot)$ represents a distribution difference metric, $\Omega(\Phi)$ is a regularizer controlling the complexity of Φ, and $\text{VAR}(\cdot)$ represents the variance of instances. This objective function aims to find a mapping function Φ that minimizes the marginal distribution difference between domains and meanwhile makes the variance of the instances as large as possible. The objective corresponding to the denominator can be optimized in several ways. One possible way is to optimize the objective of the numerator with a variance constraint. For example, the scatter matrix of the mapped instances can be enforced as an identity matrix. Another way is to optimize the objective of the numerator in a high-dimensional feature space at first. Then, a dimension reduction algorithm such as PCA or KPCA can be performed to realize the objective of the denominator.

Further, finding the explicit formulation of $\Phi(\cdot)$ is nontrivial. To solve this problem, some approaches adopt linear mapping technique or turn to the kernel trick. In general, there are three main ideas to deal with the above optimization problems.

- (Mapping Learning + Feature Extraction) A possible way is to find a high-dimensional space at first where the objectives are met by solving a kernel matrix learning problem or a transformation matrix finding problem. Then, the high-dimensional features are compacted to form a low-dimensional feature representation. For example, once the kernel matrix is learned, the principal components of the implicit high-dimensional features can be extracted to construct a new feature representation based on PCA.
- (Mapping Construction + Mapping Learning) Another way is to map the original features to a constructed high-dimensional feature space, and then a low-dimensional mapping is learned to satisfy the objective function. For example, a kernel matrix can be constructed based on a selected kernel function at first. Then, the transformation matrix can be learned, which projects the high-dimensional features into a common latent subspace.
- (Direct Low-dimensional Mapping Learning) It is usually difficult to find a desired low-dimensional mapping directly. However, if the mapping is assumed to satisfy certain conditions, it may be solvable. For example, if the low-dimensional mapping is restricted to be a linear one, the optimization problem can be easily solved.

Some approaches also attempt to match the conditional distributions and preserve the structures of the data. To achieve this, the above simple objective function needs to incorporate new terms or/and constraints. For example, the following general objective function is a possible choice:

$$\min_{\Phi} \mu\text{DIST}(X^S, X^T; \Phi) + \lambda_1\Omega^{\text{GEO}}(\Phi) + \lambda_2\Omega(\Phi)$$
$$+ (1 - \mu)\text{DIST}(Y^S|X^S, Y^T|X^T; \Phi),$$
$$s.t.\ \Phi(X)^{\text{T}}H\Phi(X) = I,\ \text{with } H = I - (\mathbf{1}/n) \in \mathbb{R}^{n \times n},$$

where μ is a parameter balancing the marginal and the conditional distribution difference [70], $\Omega^{\text{GEO}}(\Phi)$ is a regularizer controlling the geometric structure, $\Phi(X)$ is the matrix whose rows are the instances from both the source and the target domains with the extracted new feature representation, H is the centering matrix for constructing the scatter matrix, and the constraint is used to maximize the variance. The last term in the objective function denotes the measurement of the conditional distribution difference.

Before the further discussion about the above objective function, it is worth mentioning that the label information of the target-domain instances is often limited or even unknown. The lack of the label information makes it difficult to estimate the distribution difference. In order to solve this problem, some approaches resort to the pseudo-label strategy, i.e., assigning pseudo labels to the unlabeled target-domain instances. A simple method of realizing this is to train a base classifier to assign pseudo labels. By the way, there are some other methods of providing pseudo labels such as co-training [71], [72] and tri-training [73], [74]. Once the pseudo-label information is complemented, the conditional distribution difference can be measured. For example, MMD can be modified and extended to measure the conditional distribution difference. Specifically, for each label, the source-domain and the target-domain instances that belong to the same class are collected, and the estimation expression of the conditional distribution difference is given by [38]:

$$\sum_{k=1}^{|\mathcal{Y}|} \left\|\frac{1}{n_k^S}\sum_{i=1}^{n_k^S}\Phi(\mathbf{x}_i^S) - \frac{1}{n_k^T}\sum_{j=1}^{n_k^T}\Phi(\mathbf{x}_j^T)\right\|_{\mathcal{H}}^2,$$

where n_k^S and n_k^T denote the numbers of the instances in the source and the target domains with the same label $\mathcal{Y}_k$, respectively. This estimation actually measures the class-conditional distribution (i.e., $P(\mathbf{x}|y)$) difference to approximate the conditional distribution (i.e., $P(y|\mathbf{x})$) difference. Some studies improve the above estimation. For example, the work by Wang *et al.* uses a weighted method to additionally solve the class imbalance problem [70]. For better understanding, the transfer learning approaches that are the special cases of the general objective function presented in the previous paragraph are detailed as follows.

- ($\mu = 1$ and $\lambda_1 \neq 0$) The objective function of Maximum Mean Discrepancy Embedding (MMDE) is given by [75]:

$$\min_{K}\text{MMD}(X^S, X^T; \Phi) - \frac{\lambda_1}{n^S + n^T}\sum_{i \neq j}||\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)||^2$$
$$s.t.\ \forall(\mathbf{x}_i \in k\text{-NN}(\mathbf{x}_j)) \wedge (\mathbf{x}_j \in k\text{-NN}(\mathbf{x}_i)),$$
$$||\Phi(\mathbf{x}_i) - \Phi(\mathbf{x}_j)||^2 = ||\mathbf{x}_i - \mathbf{x}_j||^2, (\mathbf{x}_i, \mathbf{x}_j \in X^S \cup X^T),$$

where $k\text{-NN}(\mathbf{x})$ represents the k nearest neighbors of the instance $\mathbf{x}$. The authors design the above objective function motivated by Maximum Variance Unfolding (MVU) [76]. Instead of employing a scatter matrix constraint, the constraints and the second term of this objective function aim to maximize the distance between instances as well as preserve local geometry. The desired kernel matrix K can be learned by solving a Semi-Definite Programming (SDP) [77] problem. After obtaining the kernel matrix,

PCA is applied to it, and then the leading eigenvectors are selected to help construct a low-dimensional feature representation.

- ($\mu = 1$ and $\lambda_1 = 0$) The work by Pan *et al.* proposes an approach termed Transfer Component Analysis (TCA) [36], [78]. TCA adopts MMD to measure the marginal distribution difference and enforces the scatter matrix as the constraint. Different from MMDE that learns the kernel matrix and then further adopts PCA, TCA is a unified method that just needs to learn a linear mapping from an empirical kernel feature space to a low-dimensional feature space. In this way, it avoids solving the SDP problem, which results in relatively low computational burden. The final optimization problem can be easily solved via eigen-decomposition. TCA can also be extended to utilize the label information. In the extended version, the scatter matrix constraint is replaced by a new one that balances the label dependence (measured by HSIC) and the data variance. Besides, a graph Laplacian regularizer [30] is also added to preserve the geometry of the manifold. Similarly, the final optimization problem can also be solved by eigen-decomposition.
- ($\mu = 0.5$ and $\lambda_1 = 0$) Long *et al.* proposed an approach termed Joint Distribution Adaptation (JDA) [38]. JDA attempts to find a transformation matrix that maps the instances to a low-dimensional space where both the marginal and the conditional distribution difference are minimized. To realize it, the MMD metric and the pseudo-label strategy are adopted. The desired transformation matrix can be obtained by solving a trace optimization problem via eigen-decomposition. Further, it is obvious that the accuracy of the estimated pseudo labels affects the performance of JDA. In order to improve the labeling quality, the authors adopt the iterative refinement operations. Specifically, in each iteration, JDA is performed, and then a classifier is trained on the instances with the extracted features. Next, the pseudo labels are updated based on the trained classifier. After that, JDA is performed repeatedly with the updated pseudo labels. The iteration ends when convergence occurs. Note that JDA can be extended by utilizing the label and structure information [79], clustering information [80], various statistical and geometrical information [81], etc.
- ($\mu \in (0, 1)$ and $\lambda_1 = 0$) The paper by Wang *et al.* proposes an approach termed Balanced Distribution Adaptation (BDA) [70], which is an extension of JDA. Different from JDA which assumes that the marginal and the conditional distributions have the same importance in adaptation, BDA attempts to balance the marginal and the conditional distribution adaptation. The operations of BDA are similar to JDA. In addition, the authors also proposed the Weighted BDA (WBDA). In WBDA, the conditional distribution difference is measured by a weighted version of MMD to solve the class imbalance problem.

It is worth mentioning that some approaches transform the features into a new feature space (usually of a high dimension) and train an adaptive classifier simultaneously. To realize this, the mapping function of the features and the decision function of the classifier need to be associated. One possible way is to define the following decision function:

$f(\mathbf{x}) = \boldsymbol{\theta} \cdot \Phi(\mathbf{x}) + b$, where $\boldsymbol{\theta}$ denotes the classifier parameter; b denotes the bias. In light of the representer theorem [82], the parameter $\boldsymbol{\theta}$ can be defined as $\boldsymbol{\theta} = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i)$, and thus we have

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + b = \sum_{i=1}^{n} \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) + b,$$

where κ denotes the kernel function. By using the kernel matrix as the bridge, the regularizers designed for the mapping function can be incorporated into the classifier's objective function. In this way, the final optimization problem is usually about the parameter (e.g., α_i) or the kernel function. For example, the paper by Long *et al.* proposes a general framework termed Adaptation Regularization Based Transfer Learning (ARTL) [39]. The goals of ARTL are to learn the adaptive classifier, to minimize the structural risk, to jointly reduce the marginal and the conditional distribution difference, and to maximize the manifold consistency between the data structure and the predictive structure. The authors also proposed two specific algorithms under this framework based on different loss functions. In these two algorithms, the coefficient matrix for computing MMD and the graph Laplacian matrix for manifold regularization are constructed at first. Then, a kernel function is selected to construct the kernel matrix. After that, the classifier learning problem is converted into a parameter (i.e., α_i) solving problem, and the solution formula is also given in [39].

In ARTL, the choice of the kernel function affects the performance of the final classifier. In order to construct a robust classifier, some studies turn to kernel learning. For example, the paper by Duan *et al.* proposes a unified framework termed Domain Transfer Multiple Kernel Learning (DTMKL) [83]. In DTMKL, the kernel function is assumed to be a linear combination of a group of base kernels, i.e., $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{N} \beta_k \kappa_k(\mathbf{x}_i, \mathbf{x}_j)$. DTMKL aims to minimize the distribution difference, the classification error, etc., simultaneously. The general objective function of DTMKL can be written as follows:

$$\min_{\beta_k, f} \sigma\big(\text{MMD}(X^S, X^T; \kappa)\big) + \lambda \Omega^L(\beta_k, f),$$

where σ is any monotonically increasing function, f is the decision function with the same definition as the one in ARTL, and $\Omega^L(\beta_k, f)$ is a general term representing a group of regularizers defined on the labeled instances such as the ones for minimizing the classification error and controlling the complexity of the resultant model. The authors developed an algorithm to learn the kernel and the decision function simultaneously by using the reduced gradient descent method [84]. In each iteration, the weight coefficients of base kernels are fixed to update the decision function at first. Then, the decision function is fixed to update the weight coefficients. Note that DTMKL can incorporate many existing kernel methods. The authors proposed two specific algorithms under this framework. The first one implements the framework by using hinge loss and Support Vector Machine (SVM). The second one is an extension of the first one with an additional regularizer utilizing pseudo-label information, and the pseudo labels of the unlabeled instances are generated by using base classifiers.

4.2.4 Feature Clustering

Feature clustering aims to find a more abstract feature representation of the original features. Although it can be regarded as a way of feature extraction, it is different from the above-mentioned mapping-based extraction.

For example, some transfer learning approaches implicitly reduce the features by using the co-clustering technique, i.e., simultaneously clustering both the columns and rows of (or say, co-cluster) a contingency table based on the information theory [85]. The paper by Dai *et al.* [41] proposes an algorithm termed Co-Clustering Based Classification (CoCC), which is used for document classification. In a document classification problem, the transfer learning task is to classify the target-domain documents (represented by a document-to-word matrix) with the help of the labeled source document-to-word data. CoCC regards the co-clustering technique as a bridge to transfer the knowledge. In CoCC algorithm, both the source and the target document-to-word matrices are co-clustered. The source document-to-word matrix is co-clustered to generate the word clusters based on the known label information, and these word clusters are used as constraints during the co-clustering process of the target-domain data. The co-clustering criterion is to minimize the loss in mutual information, and the clustering results are obtained by iteration. Each iteration contains the following two steps.

1. *Document Clustering*: Each row of the target document-to-word matrix is re-ordered based on the objective function for updating the document clusters.
2. *Word Clustering*: The word clusters are adjusted to minimize the joint mutual-information loss of the source and the target document-word matrices.

After several times of iterations, the algorithm converges, and the classification results are obtained. Note that, in CoCC, the word clustering process implicitly extracts the word features to form unified word clusters.

Dai *et al.* also proposed an unsupervised clustering approach, which is termed as Self-Taught Clustering (STC) [42]. Similar to CoCC, this algorithm is also a co-clustering-based one. However, STC does not need the label information. STC aims to simultaneously co-cluster the source-domain and the target-domain instances with the assumption that these two domains share the same feature clusters in their common feature space. Therefore, two co-clustering tasks are separately performed at the same time to find the shared feature clusters. Each iteration of STC has the following steps.

1. *Instance Clustering*: The clustering results of the source-domain and the target domain instances are updated to minimize their respective loss in mutual information.
2. *Feature Clustering*: The feature clusters are updated to minimize the joint loss in mutual information.

When the algorithm converges, the clustering results of the target-domain instances are obtained.

Different from the above-mentioned co-clustering-based ones, some approaches extract the original features into concepts (or topics). In the document classification problem, the concepts represent the high-level abstractness of the words (e.g., word clusters). In order to introduce the concept-based transfer learning approaches easily, let us briefly review the Latent Semantic Analysis (LSA) [86], the Probabilistic LSA (PLSA) [87], and the Dual-PLSA [88].

- *LSA*: LSA is an approach to mapping the document-to-word matrix to a low-dimensional space (i.e., a latent semantic space) based on the SVD technique. In short, LSA attempts to find the true meanings of the words. To realize this, SVD technique is used to reduce the dimensionality, which can remove the irrelevant information and filter out the noise information from the raw data.
- *PLSA*: PLSA is developed based on a statistical view of LSA. PLSA assumes that there is a latent class variable z, which reflects the concept, associating the document d and the word w. Besides, d and w are independently conditioned on the concept z. The diagram of this graphical model is presented as follows:

$$d \xleftarrow{P(d_i|z_k)} \overset{P(z_k)}{\underset{z}{\Downarrow}} \xrightarrow{P(w_j|z_k)} w,$$

where the subscripts i, j and k represent the indexes of the document, the word, and the concept, respectively. PLSA constructs a Bayesian network, and the parameters are estimated by using the Expectation-Maximization (EM) algorithm [89].
- *Dual-PLSA*: The Dual-PLSA is an extension of PLSA. This approach assumes there are two latent variables z^d and z^w associating the documents and the words. Specifically, the variables z^d and z^w reflect the concepts behind the documents and the words, respectively. The diagram of the Dual-PLSA is provided below:

$$d \xleftarrow{P(d_i|z_{k_1}^d)} z^d \xleftarrow{P(z_{k_1}^d, z_{k_2}^w)} z^w \xrightarrow{P(w_j|z_{k_2}^w)} w.$$

The parameters of the Dual-PLSA can also be obtained based on the EM algorithm.

Some concept-based transfer learning approaches are established based on PLSA. For example, the paper by Xue *et al.* proposes a cross-domain text classification approach termed Topic-Bridged Probabilistic Latent Semantic Analysis (TPLSA) [90]. TPLSA, which is an extension of PLSA, assumes that the source-domain and the target-domain instances share the same mixing concepts of the words. Instead of performing two PLSAs for the source domain and the target domain separately, the authors merge those two PLSAs as an integrated one by using the mixing concept z as a bridge, i.e., each concept has some probabilities to produce the source-domain and the target-domain documents. The diagram of TPLSA is provided below:

$$\begin{array}{c} d^S \\ d^T \end{array} \underset{P(d_i^T|z_k)}{\overset{P(d_i^S|z_k)}{\rightleftarrows}} z \xleftarrow{P(z_k|w_j)} w.$$

Note that PLSA does not require the label information. In order to exploit the label information, the authors add the concept constraints, which include must-link and cannot-link constraints, as the penalty terms in the objective function of TPLSA. Finally, the objective function is iteratively optimized to obtain the classification results (i.e., $\arg\max_z P(z|d_i^T)$) by using the EM algorithm.

The work by Zhuang *et al.* proposes an approach termed Collaborative Dual-PLSA (CD-PLSA) for multi-domain text classification (m^S source domains and m^T target domains)

[91], [92]. CD-PLSA is an extension of Dual-PLSA. Its diagram is shown below:

$$P(\mathcal{D}_{k_0}) \quad P(d_i|z_{k_1}^d,\mathcal{D}_{k_0}) \qquad\qquad\qquad P(w_j|z_{k_2}^w,\mathcal{D}_{k_0})$$
$$\mathcal{D} \rightarrow \underset{\Downarrow}{d} \leftarrow z^d \xleftarrow{P(z_{k_1}^d,z_{k_2}^w)} z^w \rightarrow \underset{\Downarrow}{w} \quad,$$

where $1 \le k_0 \le m^S + m^T$ denotes the domain index. The domain $\mathcal{D}$ connects both the variables d and w, but is independent of the variables z^d and z^w. The label information of the source-domain instances is utilized by initializing the value $P(d_i|z_{k_1}^d,\mathcal{D}_{k_0})$ $(k_0 = 1,\cdots,m^S)$. Due to the lack of the target-domain label information, the value $P(d_i|z_{k_1}^d,\mathcal{D}_{k_0})$ $(k_0 = m^S+1,\cdots,m^S+m^T)$ can be initialized based on any supervised classifier. Similarly, the authors adopt the EM algorithm to find the parameters. Through the iterations, all the parameters in the Bayesian network are obtained. Thus, the class label of the i-th document in a target domain (denoted by $\mathcal{D}_k$) can be predicted by computing the posterior probabilities, i.e., $\arg\max_{z^d} P(z^d|d_i,\mathcal{D}_k)$.

Zhuang *et al.* further proposed a general framework that is termed as Homogeneous-Identical-Distinct-Concept Model (HIDC) [93]. This framework is also an extension of Dual-PLSA. HIDC is composed of three generative models, i.e., identical-concept, homogeneous-concept, and distinct-concept models. These three graphical models are presented below:

Identical-Concept Model: $\quad \mathcal{D} \rightarrow d \leftarrow z^d \rightarrow z_{\text{IC}}^w \rightarrow w,$

Homogeneous-Concept Model: $\quad \mathcal{D} \rightarrow d \leftarrow z^d \rightarrow z_{\text{HC}}^w \rightarrow w,$

Distinct-Concept Model: $\quad \mathcal{D} \rightarrow d \leftarrow z^d \rightarrow z_{\text{DC}}^w \rightarrow w.$

The original word concept z^w is divided into three types, i.e., z_{IC}^w, z_{HC}^w, and z_{DC}^w. In the identical-concept model, the word distributions only rely on the word concepts, and the word concepts are independent of the domains. However, in the homogeneous-concept model, the word distributions also depend on the domains. The difference between the identical and the homogeneous concepts is that z_{IC}^w is directly transferable, while z_{HC}^w is the domain-specific transferable one that may have different effects on the word distributions for different domains. In the distinct-concept model, z_{DC}^w is actually the nontransferable domain-specific one, which may only appear in a specific domain. The above-mentioned three models are combined as an integrated one, i.e., HIDC. Similar to other PLSA-related algorithms, HIDC also uses EM algorithm to get the parameters.

4.2.5 Feature Selection

Feature selection is another kind of operation for feature reduction, which is used to extract the pivot features. The pivot features are the ones that behave in the same way in different domains. Due to the stability of these features, they can be used as the bridge to transfer the knowledge. For example, Blitzer *et al.* proposed an approach termed

Structural Correspondence Learning (SCL) [94]. Briefly, SCL consists of the following steps to construct a new feature representation.

1. *Feature Selection*: SCL first performs feature selection operations to obtain the pivot features.
2. *Mapping Learning*: The pivot features are utilized to find a low-dimensional common latent feature space by using the structural learning technique [95].
3. *Feature Stacking*: A new feature representation is constructed by feature augmentation, i.e., stacking the original features with the obtained low-dimensional features.

Take the part-of-speech tagging problem as an example. The selected pivot features should occur frequently in source and target domains. Therefore, determiners can be included in pivot features. Once all the pivot features are defined and selected, a number of binary linear classifiers whose function is to predict the occurrence of each pivot feature are constructed. Without losing generality, the decision function of the i-th classifier, which is used to predict the i-th pivot feature, can be formulated as $f_i(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}_i \cdot \mathbf{x})$, where $\mathbf{x}$ is assumed to be a binary feature input. And the i-th classifier is trained on all the instances excluding the features derived from the i-th pivot feature. The following formula can be used to estimate the i-th classifier's parameters, i.e.,

$$\boldsymbol{\theta}_i = \arg\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{j=1}^{n} \mathcal{L}\left(\boldsymbol{\theta}\cdot\mathbf{x}_j, \text{Row}_i(\mathbf{x}_j)\right) + \lambda||\boldsymbol{\theta}||^2,$$

where $\text{Row}_i(\mathbf{x}_j)$ denotes the true value of the unlabeled instance $\mathbf{x}_j$ in terms of the i-th pivot feature. By stacking the obtained parameter vectors as column elements, a matrix $\tilde{W}$ is obtained. Next, based on singular value decomposition (SVD), the top-k left singular vectors, which are the principal components of the matrix $\tilde{W}$, are taken to construct the transformation matrix W. At last, the final classifier is trained on the labeled instances in an augmented feature space, i.e., $([\mathbf{x}_i^L; W^{\text{T}}\mathbf{x}_i^L]^{\text{T}}, y_i^L)$.

4.2.6 Feature Encoding

In addition to feature extraction and selection, feature encoding is also an effective tool. For example, autoencoders, which are often adopted in deep learning area, can be used for feature encoding. An autoencoder consists of an encoder and a decoder. The encoder tries to produce a more abstract representation of the input, while the decoder aims to map back that representation and to minimize the reconstruction error. Autoencoders can be stacked to build a deep learning architecture. Once an autoencoder completes the training process, another autoencoder can be stacked at the top of it. The newly added autoencoder is then trained by using the encoded output of the upper-level autoencoder as its input. In this way, deep learning architectures can thus be constructed.

Some transfer learning approaches are developed based on autoencoders. For example, the paper by Glorot *et al.* proposes an approach termed Stacked Denoising Autoencoder (SDA) [96]. The denoising autoencoder, which can enhance the robustness, is an extension of the basic one [97]. This kind of autoencoder contains a randomly corrupting mechanism that adds noise to the input before mapping. For

example, an input can be corrupted or partially destroyed by adding a masking noise or Gaussian noise. The denoising autoencoder is then trained to minimize the denoising reconstruction error between the original clean input and the output. The SDA algorithm proposed in the paper mainly encompasses the following steps.

1. *Autoencoder Training*: The source-domain and target-domain instances are used to train a stack of denoising autoencoders in a greedy layer-by-layer way.
2. *Feature Encoding & Stacking*: A new feature representation is constructed by stacking the encoding output of intermediate layers, and the features of the instances are transformed into the obtained new representation.
3. *Learner Training*: The target classifier is trained on the transformed labeled instances.

Although the SDA algorithm has excellent performance for feature extraction, it still has some drawbacks such as high computational and parameter-estimation cost. In order to shorten the training time and to speed up traditional SDA algorithms, Chen *et al.* proposed a modified version of SDA, i.e., Marginalized Stacked Linear Denoising Autoencoder (mSLDA) [98], [99]. This algorithm adopts linear autoencoders and marginalizes the randomly corrupting step in a closed form. It may seem that linear autoencoders are too simple to learn complex features. However, the authors observe that linear autoencoders are often sufficient to achieve competent performance when encountering high dimensional data. The basic architecture of mSLDA is a single-layer linear autoencoder. The corresponding single-layer mapping matrix W (augmented with a bias column for convenience) should minimize the expected squared reconstruction loss function, i.e.,

$$W = \arg\min_{W} \frac{1}{2n} \sum_{i=1}^{n} \mathbb{E}_{P(\tilde{\mathbf{x}}_i|\mathbf{x})} \left[||\mathbf{x}_i - W\tilde{\mathbf{x}}_i||^2 \right],$$

where $\tilde{\mathbf{x}}_i$ denotes the corrupted version of the input $\mathbf{x}_i$. The solution of W is given by [98], [99]:

$$W = \left(\sum_{i=1}^{n} \mathbf{x}_i \mathbb{E}[\tilde{\mathbf{x}}_i]^{\mathrm{T}} \right) \left(\sum_{i=1}^{n} \mathbb{E}\left[\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^{\mathrm{T}} \right] \right)^{-1}.$$

When the corruption strategy is determined, the above formulas can be further expanded and simplified into a specific form. Note that, in order to insert nonlinearity, a nonlinear function is used to squash the output of each autoencoder after we obtain the matrix W in a closed form. Then, the next linear autoencoder is stacked to the current one in a similar way to SDA. In order to deal with high dimensional data, the authors also put forward an extension approach to further reduce the computational complexity.

4.2.7 Feature Alignment

Note that feature augmentation and feature reduction mainly focus on the explicit features in a feature space. In contrast, in addition to the explicit features, feature alignment also focuses on some implicit features such as the statistic features and the spectral features. Therefore, feature alignment can play various roles in the feature transformation process. For example, the explicit features can be aligned to generate a new feature representation, or

the implicit features can be aligned to construct a satisfied feature transformation.

There are several kinds of features that can be aligned, which includes subspace features, spectral features, and statistic features. Take the subspace feature alignment as an example. A typical approach mainly has the following steps.

1. *Subspace Generation*: In this step, the instances are used to generate the respective subspaces for the source and the target domains. The orthonormal bases of the source and the target domain subspaces are then obtained, which are denoted by M_S and M_T, respectively. These bases are used to learn the shift between the subspaces.
2. *Subspace Alignment*: In the second step, a mapping, which aligns the bases M_S and M_T of the subspaces, is learned. And the features of the instances are projected to the aligned subspaces to generate new feature representation.
3. *Learner Training*: Finally, the target learner is trained on the transformed instances.

For example, the work by Fernando *et al.* proposes an approach termed Subspace Alignment (SA) [100]. In SA, the subspaces are generated by performing PCA; the bases M_S and M_T are obtained by selecting the leading eigenvectors. Then, a transformation matrix W is learned to align the subspaces, which is given by [100]:

$$W = \arg\min_{W} ||M_S W - M_T||_F^2 = M_S^{\mathrm{T}} M_T,$$

where $|| \cdot ||_F$ denotes the Frobenius norm. Note that the matrix W aligns M_S with M_T, or say, transforms the source subspace coordinate system into the target subspace coordinate system. The transformed low-dimensional source-domain and target-domain instances are given by $X^S M_S W$ and $X^T M_T$, respectively. Finally, a learner can be trained on the resultant transformed instances.

In light of SA, a number of transfer learning approaches are established. For example, the paper by Sun and Saenko proposes an approach that aligns both the subspace bases and the distributions [101], which is termed as Subspace Distribution Alignment between Two Subspaces (SDA-TS). In SDA-TS, the transformation matrix W is formulated as $W = M_S^{\mathrm{T}} M_T Q$, where Q is a matrix used to align the distribution difference. The transformation matrix W in SA is a special case of the one in SDA-TS by setting Q to an identity matrix. Note that SA is a symmetrical feature-based approach, while SDA-TS is an asymmetrical one. In SDA-TS, the labeled source-domain instances are projected to the source subspace, then mapped to the target subspace, and finally mapped back to the target domain. The transformed source-domain instances are formulated as $X^S M_S W M_T^{\mathrm{T}}$.

Another representative subspace feature alignment approach is Geodesic Flow Kernel (GFK) [102], which is proposed by Gong *et al*. GFK is closely related to a previous approach termed Geodesic Flow Subspaces (GFS) [103]. Before introducing GFK, let us review the steps of GFS at first. GFS is inspired by incremental learning. Intuitively, utilizing the information conveyed by the potential path between two domains may be beneficial to the domain adaptation. GFS generally takes the following steps to align features.

1. *Subspace Generation*: GFS first generates two subspaces of the source and the target domains by performing PCA, respectively.
2. *Subspace Interpolation*: The two obtained subspaces can be viewed as two points on the Grassmann manifold [104]. A finite number of the interpolated subspaces are generated between these two subspaces based on the geometric properties of the manifold.
3. *Feature Projection & Stacking*: The original features are transformed by stacking the corresponding projections from all the obtained subspaces.

Despite the usefulness and superiority of GFS, there is a problem about how to determine the number of the interpolated subspaces. GFK resolves this problem by integrating infinite number of the subspaces located on the geodesic curve from the source subspace to the target one. The key of GFK is to construct an infinite-dimensional feature space that incorporating the information of all the subspaces lying on the geodesic flow. In order to compute the inner product in the resultant infinite-dimensional space, the geodesic-flow kernel is defined and derived. In addition, a subspace-disagreement measure is proposed to select the optimal dimensionality of the subspaces; a rank-of-domain metric is also proposed to select the optimal source domain when multi-source domains are available.

Statistic feature alignment is another kind of feature alignment. For example, Sun *et al.* proposed an approach termed Co-Relation Alignment (CORAL) [105]. CORAL constructs the transformation matrix of the source features by aligning the second-order statistic features, i.e., the covariance matrices. The transformation matrix W is given by [105]:

$$W = \arg\min_{W} ||W^{\mathrm{T}} C_S W - C_T||_F^2,$$

where C denotes the covariance matrix. Note that, compared to the above subspace-based approaches, CORAL avoids subspace generation as well as projection and is very easy to implement.

Some transfer learning approaches are established based on spectral feature alignment. In traditional machine learning area, spectral clustering is a clustering technique based on graph theory. The key of this technique is to utilize the spectrum, i.e., eigenvalues, of the similarity matrix to reduce the dimension of the features before clustering. The similarity matrix is constructed to quantitatively assess the relative similarity of each pair of data/vertices. On the basis of spectral clustering and feature alignment, Spectral Feature Alignment (SFA) [106] is proposed by Pan *et al.* SFA is an algorithm for sentiment classification. This algorithm tries to identify the domain-specific words and domain-independent words in different domains, and then aligns these domain-specific word features to construct a low-dimensional feature representation. SFA generally contains the following five steps.

1. *Feature Selection*: In this step, feature selection operations are performed to select the domain-independent/pivot features. The paper presents three strategies to select domain-independent features. These strategies are based on the occurrence frequency of words, the mutual information between features and labels [107], and the mutual information between features and domains, respectively.
2. *Similarity Matrix Construction*: Once the domain-specific and the domain-independent features are identified, a bipartite graph is constructed. Each edge of this bipartite graph is assigned with a weight that measures the co-occurrence relationship between a domain-specific word and a domain-independent word. Based on the bipartite graph, a similarity matrix is then constructed.
3. *Spectral Feature Alignment*: In this step, a spectral clustering algorithm is adapted and performed to align domain-specific features [108], [109]. Specifically, based on the eigenvectors of the graph Laplacian, a feature alignment mapping is constructed, and the domain-specific features are mapped into a low-dimensional feature space.
4. *Feature Stacking*: The original features and the low-dimensional features are stacked to produce the final feature representation.
5. *Learner Training*: The target learner is trained on the labeled instances with the final feature representation.

There are some other spectral transfer learning approaches. For example, the work by Ling *et al.* proposes an approach termed Cross-Domain Spectral Classifier (CDSC) [110]. The general ideas and steps of this approach are presented as follows.

1. *Similarity Matrix Construction*: In the first step, two similarity matrices are constructed corresponding to the whole instances and the target-domain instances, respectively.
2. *Spectral Feature Alignment*: An objective function is designed with respect to a graph-partition indicator vector; a constraint matrix is constructed, which contains pair-wise must-link information. Instead of seeking the discrete solution of the indicator vector, the solution is relaxed to be continuous, and the eigen-system problem corresponding to the objective function is solved to construct the aligned spectral features [111].
3. *Learner Training*: A traditional classifier is trained on the transformed instances.

To be more specific, the objective function has a form of the generalized Rayleigh quotient, which aims to find the optimal graph partition that respects the label information with small cut-size [112], to maximize the separation of the target-domain instances, and to fit the constraints of the pair-wise property. After eigen-decomposition, the last eigenvectors are selected and combined as a matrix, and then the matrix is normalized. Each row of the normalized matrix represents a transformed instance.

5 MODEL-BASED INTERPRETATION

Transfer learning approaches can also be interpreted from the model perspective. Fig. 4 shows the corresponding strategies and the objectives. The main objective of a transfer learning model is to make accurate prediction results on the target domain, e.g., classification or clustering results. Note that a transfer learning model may consist of a few sub-modules such as classifiers, extractors, or encoders. These sub-modules may play different roles, e.g., feature adaptation or pseudo label generation. In this section, some related
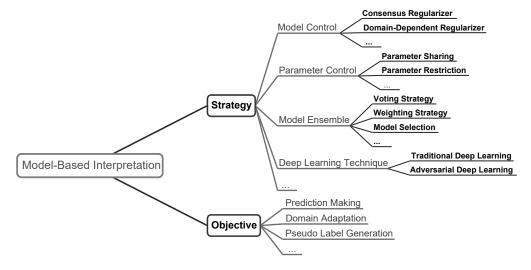
Fig. 4. Strategies and objectives of the transfer learning approaches from the model perspective.

transfer learning approaches are introduced in proper order according to the strategies shown in Fig. 4.

5.1 Model Control Strategy

From the perspective of model, a natural thought is to directly add the model-level regularizers to the learner's objective function. In this way, the knowledge contained in the pre-obtained source models can be transferred into the target model during the training process. For example, the paper by Duan *et al.* proposes a general framework termed Domain Adaptation Machine (DAM) [113], [114], which is designed for multi-source transfer learning. The goal of DAM is to construct a robust classifier for the target domain with the help of some pre-obtained base classifiers that are respectively trained on multiple source domains. The objective function is given by:

$$\min_{f^T} \mathcal{L}^{T,L}(f^T) + \lambda_1 \Omega^{\mathrm{D}}(f^T) + \lambda_2 \Omega(f^T),$$

where the first term represents the loss function used to minimize the classification error of the labeled target-domain instances, the second term denotes different regularizers, and the third term is used to control the complexity of the final decision function f^T. Different types of the loss functions can be adopted in $\mathcal{L}^{T,L}(f^T)$ such as the square error or the cross-entropy loss. Some transfer learning approaches can be regarded as the special cases of this framework to some extent.

- (Consensus Regularizer) The work by Luo *et al.* proposes a framework termed Consensus Regularization Framework (CRF) [115], [116]. CRF is designed for multi-source transfer learning with no labeled target-domain instances. The framework constructs m^S classifiers corresponding to each source domain, and these classifiers are required to reach mutual consensuses on the target domain. The objective function of each source classifier, denoted by f_k^S

(with $k = 1, \cdots, m^S$), is similar to that of DAM, which is presented below:

$$\min_{f_k^S} - \sum_{i=1}^{n^{S_k}} \log P(y_i^{S_k}|\mathbf{x}_i^{S_k}; f_k^S) + \lambda_2 \Omega(f_k^S)$$
$$+ \lambda_1 \sum_{i=1}^{n^{T,U}} \sum_{y_j \in \mathcal{Y}} S\Big(\frac{1}{m^S} \sum_{k_0=1}^{m^S} P(y_j|\mathbf{x}_i^{T,U}; f_{k_0}^S)\Big),$$

where f_k^S denotes the decision function corresponding to the k-th source domain, and $S(x) = -x \log x$. The first term is used to quantify the classification error of the k-th classifier on the k-th source domain, and the last term is the consensus regularizer in the form of cross-entropy. The consensus regularizer can not only enhance the agreement of all the classifiers, but also reduce the uncertainty of the predictions on the target domain. The authors implement this framework based on the logistic regression. A difference between DAM and CRF is that DAM explicitly constructs the target classifier, while CRF makes the target predictions based on the reached consensus from the source classifiers.

- (Domain-dependent Regularizer) Fast-DAM is a specific algorithm of DAM [113]. In light of the manifold assumption [30] and the graph-based regularizer [117], [118], Fast-DAM designs a domain-dependent regularizer. The objective function is given by:

$$\min_{f^T} \sum_{j=1}^{n^{T,L}} \Big(f^T(\mathbf{x}_j^{T,L}) - y_j^{T,L} \Big)^2 + \lambda_2 \Omega(f^T)$$
$$+ \lambda_1 \sum_{k=1}^{m^S} \beta_k \sum_{i=1}^{n^{T,U}} \Big(f^T(\mathbf{x}_i^{T,U}) - f_k^S(\mathbf{x}_i^{T,U}) \Big)^2,$$

where f_k^S ($k = 1, 2, \cdots, m^S$) denotes the pre-obtained source decision function for the k-th source domain and β_k represents the weighting parameter that is determined by the relevance between the target domain and the k-th source domain and can be measured based on the MMD metric. The third term is the domain-dependent regularizer, which transfers the knowledge contained in

the source classifier motivated by domain dependence. In [113], the authors also introduce and add a new term to the above objective function based on ε-insensitive loss function [119], which makes the resultant model have high computational efficiency.

- (Domain-dependent Regularizer + Universum Regularizer) Univer-DAM is an extension of the Fast-DAM [114]. Its objective function contains an additional regularizer, i.e., Universum regularizer. This regularizer usually utilizes an additional dataset termed Universum where the instances do not belong to either the positive or the negative class [120]. The authors treat the source-domain instances as the Universum for the target domain, and the objective function of Univer-DAM is presented as follows:

$$\min_{f^T} \sum_{j=1}^{n^{T,L}} \left(f^T(\mathbf{x}_j^{T,L}) - y_j^{T,L} \right)^2 + \lambda_2 \sum_{j=1}^{n^S} \left(f^T(\mathbf{x}_j^S) \right)^2$$
$$+ \lambda_1 \sum_{k=1}^{m^S} \beta_k \sum_{i=1}^{n^{T,U}} \left(f^T(\mathbf{x}_i^{T,U}) - f_k^S(\mathbf{x}_i^{T,U}) \right)^2 + \lambda_3 \Omega(f^T).$$

Similar to Fast-DAM, the ε-insensitive loss function can also be utilized [114].

5.2 Parameter Control Strategy

The parameter control strategy focuses on the parameters of models. For example, in the application of object categorization, the knowledge from known source categories can be transferred into target categories via object attributes such as shape and color [121]. The attribute priors, i.e., probabilistic distribution parameters of the image features corresponding to each attribute, can be learned from the source domain and then used to facilitate learning the target classifier. The parameters of a model actually reflect the knowledge learned by the model. Therefore, it is possible to transfer the knowledge at the parametric level.

5.2.1 Parameter Sharing

An intuitive way of controlling the parameters is to directly share the parameters of the source learner to the target learner. Parameter sharing is widely employed especially in the network-based approaches. For example, if we have a neural network for the source task, we can freeze (or say, share) most of its layers and only finetune the last few layers to produce a target network. The network-based approaches are introduced in Section 5.4.

In addition to network-based parameter sharing, matrix-factorization-based parameter sharing is also workable. For example, Zhuang et al. proposed an approach for text classification, which is referred to as Matrix Tri-Factorization Based Classification Framework (MTrick) [122]. The authors observe that, in different domains, different words or phrases sometimes express the same or similar connotative meaning. Thus, it is more effective to use the concepts behind the words rather than the words themselves as a bridge to transfer the knowledge in source domains. Different from PLSA-based transfer learning approaches that utilize the concepts by constructing Bayesian networks, MTrick attempts to find the connections between the document classes and the concepts conveyed by the word clusters

through matrix tri-factorization. These connections are considered to be the stable knowledge that is supposed to be transferred. The main idea is to decompose a document-to-word matrix into three matrices, i.e., document-to-cluster, connection, and cluster-to-word matrices. Specifically, by performing the matrix tri-factorization operations on the source and the target document-to-word matrices respectively, a joint optimization problem is constructed, which is given by

$$\min_{Q,R,W} ||X^S - Q^S R W^S||^2 + \lambda_1 ||X^T - Q^T R W^T||^2$$
$$+ \lambda_2 ||Q^S - \check{Q}^S||^2$$
$$s.t. \text{ Normalization Constraints,}$$

where X denotes the document-to-word matrix, Q denotes the document-to-cluster matrix, R represents the transformation matrix from document clusters to word clusters, W denotes the cluster-to-word matrix, n^d denotes the number of the documents, and $\check{Q}^S$ represents the label matrix. The matrix $\check{Q}^S$ is constructed based on the class information of the source-domain documents. If the i-th document belongs to the k-th class, $\check{Q}^S_{[i,k]} = 1$. In the above objective function, the matrix R is actually the shared parameter. The first term aims to tri-factorize the source document-to-word matrix, and the second term decomposes the target document-to-word matrix. The last term incorporates the source-domain label information. The optimization problem is solved based on the alternating iteration method. Once the solution of Q^T is obtained, the class index of the k-th target-domain instance is the one with the maximum value in the k-th row of Q^T.

Further, Zhuang et al. extended MTrick and proposed an approach termed Triplex Transfer Learning (TriTL) [123]. MTrick assumes that the domains share the similar concepts behind their word clusters. In contrast, TriTL assumes that the concepts of these domains can be further divided into three types, i.e., domain-independent, transferable domain-specific, and nontransferable domain-specific concepts, which is similar to HIDC. This idea is motivated by Dual Transfer Learning (DTL), where the concepts are assumed to be composed of the domain-independent ones and the transferable domain-specific ones [124]. The objective function of TriTL is provided as follows:

$$\min_{Q,R,W} \sum_{k=1}^{m^S + m^T} ||X_k - Q_k \begin{bmatrix} R^{\text{DI}} & R^{\text{TD}} & R_k^{\text{ND}} \end{bmatrix} \begin{bmatrix} W^{\text{DI}} \\ W_k^{\text{TD}} \\ W_k^{\text{ND}} \end{bmatrix} ||^2$$
$$s.t. \text{ Normalization Constraints,}$$

where the definitions of the symbols are similar to those of MTrick and the subscript k denotes the index of the domains with the assumption that the first m^S domains are the source domains and the last m^T domains are the target domains. The authors proposed an iterative algorithm to solve the optimization problem. And in the initialization phase, W^{DI} and W_k^{TD} are initialized based on the clustering results of the PLSA algorithm, while W_k^{UT} is randomly initialized; the PLSA algorithm is performed on the combination of the instances from all the domains.

There are some other approaches developed based on matrix factorization. Wang et al. proposed a transfer learning framework for image classification [125]. Wang et al.

proposed a softly associative approach that integrates two matrix tri-factorizations into a joint framework [126]. Do *et al.* utilized matrix tri-factorization to discover both the implicit and the explicit similarities for cross-domain recommendation [127].

5.2.2 Parameter Restriction

Another parameter-control-type strategy is to restrict the parameters. Different from the parameter sharing strategy that enforces the models share some parameters, parameter restriction strategy only requires the parameters of the source and the target models to be similar.

Take the approaches to category learning as examples. The category-learning problem is to learn a new decision function for predicting a new category (denoted by the $(k + 1)$-th category) with only limited target-domain instances and k pre-obtained binary decision functions. The function of these pre-obtained decision functions is to predict which of the k categories an instance belongs to. In order to solve the category-learning problem, Tommasi *et al.* proposed an approach termed Single-Model Knowledge Transfer (SMKL) [128]. SMKL is based on Least-Squares SVM (LS-SVM). The advantage of LS-SVM is that LS-SVM transforms inequality constraints to equality constraints and has high computational efficiency; its optimization is equivalent to solving a linear equation system problem instead of a quadratic programming problem. SMKL selects one of the pre-obtained binary decision functions, and transfers the knowledge contained in its parameters. The objective function is given by

$$\min_f \frac{1}{2} \left\| \boldsymbol{\theta} - \beta \tilde{\boldsymbol{\theta}} \right\|^2 + \frac{\lambda}{2} \sum_{j=1}^{n^{T,L}} \eta_j \left(f(\mathbf{x}_j^{T,L}) - y_j^{T,L} \right)^2,$$

where $f(\mathbf{x}) = \boldsymbol{\theta} \cdot \Phi(\mathbf{x}) + b$, β is the weighting parameter controlling the transfer degree, $\tilde{\boldsymbol{\theta}}$ is the parameter of a selected pre-obtained model, and η_j is the coefficient for resolving the label imbalance problem. The kernel parameter and the tradeoff parameter are chosen based on cross-validation. In order to find the optimal weighting parameter, the authors refer to an earlier work [129]. In [129], Cawley proposed a model selection mechanism for LS-SVM, which is based on the leave-one-out cross-validation method. The superiority of this method is that the leave-one-out error for each instance can be obtained in a closed form without performing the real cross-validation experiment. Motivated by Cawley's work, the generalization error can be easily estimated to guide the parameter setting in SMKL.

Tommasi *et al.* further extended SMKL by utilizing all the pre-obtained decision functions. In [130], an approach that is referred to as Multi-Model Knowledge Transfer (MMKL) is proposed. Its objective function is presented as follows:

$$\min_f \frac{1}{2} \left\| \boldsymbol{\theta} - \sum_{i=1}^{k} \beta_i \boldsymbol{\theta}_i \right\|^2 + \frac{\lambda}{2} \sum_{j=1}^{n^{T,L}} \eta_j \left(f(\mathbf{x}_j^{T,L}) - y_j^{T,L} \right)^2,$$

where $\boldsymbol{\theta}_i$ and β_i are the model parameter and the weighting parameter of the i-th pre-obtained decision function, respectively. The leave-one-out error can also be obtained in a closed form, and the optimal value of β_i ($i = 1, 2, \cdots, k$) is the one that maximizes the generalization performance.

5.3 Model Ensemble Strategy

In sentiment analysis applications related to product reviews, data or models from multiple product domains are available and can be used as the source domains [131]. Combining data or models directly into a single domain may not be successful because the distributions of these domains are different from each other. Model ensemble is another commonly used strategy. This strategy aims to combine a number of weak classifiers to make the final predictions. Some previously mentioned transfer learning approaches already adopted this strategy. For example, TrAdaBoost and MsTrAdaBoost ensemble the weak classifiers via voting and weighting, respectively. In this subsection, several typical ensemble-based transfer learning approaches are introduced to help readers better understand the function and the appliance of this strategy.

As mentioned in Section 4.1, TaskTrAdaBoost, which is an extension of TrAdaBoost for handling multi-source scenarios, is proposed in the paper [33]. TaskTrAdaBoost mainly has the following two stages.

1. *Candidate Classifier Construction*: In the first stage, a group of candidate classifiers are constructed by performing AdaBoost on each source domain. Note that, for each source domain, each iteration of AdaBoost results in a new weak classifier. In order to avoid the overfitting problem, the authors introduced a threshold to pick the suitable classifiers into the candidate group.
2. *Classifier Selection and Ensemble*: In the second stage, a revised version of AdaBoost is performed on the target-domain instances to construct the final classifier. In each iteration, an optimal candidate classifier which has the lowest classification error on the labeled target-domain instances is picked out and assigned with a weight based on the classification error. Then, the weight of each target-domain instance is updated based on the performance of the selected classifier on the target domain. After the iteration process, the selected classifiers are ensembled to produce the final predictions.

The difference between the original AdaBoost and the second stage of TaskTrAdaBoost is that, in each iteration, the former constructs a new candidate classifier on the weighted target-domain instances, while the latter selects one pre-obtained candidate classifier which has the minimal classification error on the weighted target-domain instances.

The paper by Gao *et al.* proposes another ensemble-based framework that is referred to as Locally Weighted Ensemble (LWE) [132]. LWE focuses on the ensemble process of various learners; these learners could be constructed on different source domains, or be built by performing different learning algorithms on a single source domain. Different from TaskTrAdaBoost that learns the global weight of each learner, the authors adopted the local-weight strategy, i.e., assigning adaptive weights to the learners based on the local manifold structure of the target-domain test set. In LWE, a learner is usually assigned with different weights when classifying different target-domain instances. Specifically, the authors adopt a graph-based approach to estimate the weights. The steps for weighting are outlined below.

1. *Graph Construction*: For the i-th source learner, a graph $G_{S_i}^T$ is constructed by using the learner to classify the

target-domain instances in the test set; if two instances are classified into the same class, they are connected in the graph. Another graph G^T is constructed for the target-domain instances as well by performing a clustering algorithm.

2. *Learner Weighting*: The weight of the i-th learner for the j-th target-domain instance $\mathbf{x}_j^T$ is proportional to the similarity between the instance's local structures in $G_{S_i}^T$ and G^T. And the similarity can be measured by the percentage of the common neighbors of $\mathbf{x}_j^T$ in these two graphs.

Note that this weighting scheme is based on the clustering-manifold assumption, i.e., if two instances are close to each other in a high-density region, they often have similar labels. In order to check the validity of this assumption for the task, the target task is tested on the source-domain training set(s). Specifically, the clustering quality of the training set(s) is quantified and checked by using a metric such as purity or entropy. If the clustering quality is not satisfactory, uniform weights are assigned to the learners instead. Besides, it is intuitive that if the measured structure similarity is particularly low for every learner, weighting and combining these learners seems unwise. Therefore, the authors introduce a threshold and compare it to the average similarity. If the similarity is lower than the threshold, the label of $\mathbf{x}_j^T$ is determined by the voting scheme among its reliable neighbors, where the reliable neighbors are the ones whose label predictions are made by the combined classifier.

The above-mentioned TaskTrAdaBoost and LWE approaches mainly focus on the ensemble process. In contrast, some studies focus more on the construction of weak learners. For example, Ensemble Framework of Anchor Adapters (ENCHOR) [133] is a weighting ensemble framework proposed by Zhuang *et al.* An anchor is a specific instance. Different from TrAdaBoost which adjusts weights of instances to train and produce a new learner iteratively, ENCHOR constructs a group of weak learners via using different representations of the instances produced by anchors. The thought is that the higher similarity between a certain instance and an anchor, the more likely the feature of that instance remains unchanged relative to the anchor, where the similarity can be measured by using the cosine or Gaussian distance function. ENCHOR contains the following steps.

1. *Anchor Selection*: In this step, a group of anchors are selected. These anchors can be selected based on some rules or even randomly. In order to improve the final performance of ENCHOR, the authors proposed a method of selecting high-quality anchors [133].
2. *Anchor-based Representation Generation*: For each anchor and each instance, the feature vector of an instance is directly multiplied by a coefficient that measures the distance from the instance to the anchor. In this way, each anchor produces a new pair of anchor-adapted source and target instance sets.
3. *Learner Training and Ensemble*: The obtained pairs of instance sets can be respectively used to train learners. Then, the resultant learners are weighted and combined to make the final predictions.

The framework ENCHOR is easy to be realized in a parallel manner in that the operations performed on each anchor are independent.

5.4 Deep Learning Technique

Deep learning methods are particularly popular in the field of machine learning. Many researchers utilize the deep learning techniques to construct transfer learning models. For example, the SDA and the mSLDA approaches mentioned in Section 4.2.6 utilize the deep learning techniques. In this subsection, we specifically discuss the deep-learning-related transfer learning models. The deep learning approaches introduced are divided into two types, i.e., non-adversarial (or say, traditional) ones and adversarial ones.

5.4.1 Traditional Deep Learning

As said earlier, autoencoders are often used in deep learning area. In addition to SDA and mSLDA, there are some other reconstruction-based transfer learning approaches. For example, the paper by Zhuang *et al.* proposes an approach termed Transfer Learning with Deep Autoencoders (TLDA) [44], [134]. TLDA adopts two autoencoders for the source and the target domains, respectively. These two autoencoders share the same parameters. The encoder and the decoder both have two layers with activation functions. The diagram of the two autoencoders is presented as follows:

$$X^S \xrightarrow{(W_1,b_1)} Q^S \xrightarrow[\text{Softmax Regression}]{(W_2,b_2)} R^S \xrightarrow{(\hat{W}_2,\hat{b}_2)} \tilde{Q}^S \xrightarrow{(\hat{W}_1,\hat{b}_1)} \tilde{X}^S,$$

$$\Uparrow$$
$$\text{KL Divergence}$$
$$\Downarrow$$

$$X^T \xrightarrow{(W_1,b_1)} Q^T \xrightarrow[\text{Softmax Regression}]{(W_2,b_2)} R^T \xrightarrow{(\hat{W}_2,\hat{b}_2)} \tilde{Q}^T \xrightarrow{(\hat{W}_1,\hat{b}_1)} \tilde{X}^T.$$

There are several objectives of TLDA, which are listed as follows.

1. *Reconstruction Error Minimization*: The output of the decoder should be extremely close to the input of encoder. In other words, the distance between X^S and $\tilde{X}^S$ as well as the distance between X^T and $\tilde{X}^T$ should be minimized.
2. *Distribution Adaptation*: The distribution difference between Q^S and Q^T should be minimized.
3. *Regression Error Minimization*: The output of the encoder on the labeled source-domain instances, i.e., R^S, should be consistent with the corresponding label information Y^S.

Therefore, the objective function of TLDA is given by

$$\min_{\Theta} \ \mathcal{L}_{\text{REC}}(X,\tilde{X}) + \lambda_1 \text{KL}(Q^S||Q^T) + \lambda_2 \Omega(W,b,\hat{W},\hat{b})$$
$$+ \lambda_3 \mathcal{L}_{\text{REG}}(R^S,Y^S),$$

where the first term represents the reconstruction error, $\text{KL}(\cdot)$ represents the KL divergence, the third term controls the complexity, and the last term represents the regression error. TLDA is trained by using a gradient descent method. The final predictions can be made in two different ways. The first way is to directly use the output of the encoder to make predictions. And the second way is to treat the autoencoder as a feature extractor, and then train the target classifier on the labeled instances with the feature representation produced by the encoder's first-layer output.

In addition to the reconstruction-based domain adaptation, discrepancy-based domain adaptation is also a popular direction. In earlier research, the shallow neural networks are tried to learn the domain-independent feature representation [135]. It is found that the shallow architectures often make it difficult for the resultant models to achieve excellent performance. Therefore, many studies turn to utilize deep neural networks. Tzeng *et al.* [136] added a single adaptation layer and a discrepancy loss to the deep neural network, which improves the performance. Further, Long *et al.* performed multi-layer adaptation and utilized multi-kernel technique, and they proposed an architecture termed Deep Adaptation Networks (DAN) [137].

For better understanding, let us review DAN in detail. DAN is based on AlexNet [138] and its architecture is presented below [137].

$$X^S \xrightarrow[\text{1st}]{\text{conv}} Q_1^S \xrightarrow[\text{conv}]{} Q_5^S \nearrow$$

In the above network, the features are first extracted by five convolutional layers in a general-to-specific manner. Next, the extracted features are fed into one of the two fully connected networks switched by their original domains. These two networks both consist of three fully connected layers that are specialized for the source and the target domains. DAN has the following objectives.

1. *Classification Error Minimization*: The classification error of the labeled instances should be minimized. The cross-entropy loss function is adopted to measure the prediction error of the labeled instances.
2. *Distribution Adaptation*: Multiple layers, which include the representation layers and the output layer, can be jointly adapted in a layer-wise manner. Instead of using the single-kernel MMD to measure the distribution difference, the authors turn to MK-MMD. The authors adopt the linear-time unbiased estimation of MK-MMD to avoid numerous inner product operations [62].
3. *Kernel Parameter Optimization*: The weighting parameters of the multiple kernels in MK-MMD should be optimized to maximize the test power [62].

The objective function of the DAN network is given by:

$$\min_{\Theta} \max_{\kappa} \sum_{i=1}^{n^L} \mathcal{L}\left(f(\mathbf{x}_i^L), y_i^L\right) + \lambda \sum_{l=6}^{8} \text{MK-MMD}(R_l^S, R_l^T; \kappa),$$

where l denotes the index of the layer. The above optimization is actually a minimax optimization problem. The maximization of the objective function with respect to the kernel function κ aims to maximize the test power. After this step, the subtle difference between the source and the target domains are magnified. This train of thought is similar to the Generative Adversarial Network (GAN) [139]. In the training process, the DAN network is initialized by a pre-trained AlexNet [138]. There are two categories of parameters that should be learned, i.e., the network parameters

and the weighting parameters of the multiple kernels. Given that the first three convolutional layers output the general features and are transferable, the authors freeze them and fine-turn the last two convolutional layers and the two fully connected layers [140]. The last fully connected layer (or say, the classifier layer) is trained from scratch.

Long *et al.* further extended the above DAN approach and proposed the DAN framework [141]. The new characteristics are summarized as follows.

1. *Regularizer Adding*: The framework introduces an additional regularizer to minimize the uncertainty of the predicted labels of the unlabeled target-domain instances, which is motivated by entropy minimization criterion [142].
2. *Architecture Generalizing*: The DAN framework can be applied to many other architectures such as GoogLeNet [143] and ResNet [144].
3. *Measurement Generalizing*: The distribution difference can be estimated by other metrics. For example, in addition to MK-MMD, the authors also present the Mean Embedding test for distribution adaptation [145].

The objective function of the DAN framework is given by:

$$\min_{\Theta} \max_{\kappa} \sum_{i=1}^{n^L} \mathcal{L}\left(f(\mathbf{x}_i^L), y_i^L\right) + \lambda_1 \sum_{l=l_{\text{strt}}}^{l_{\text{end}}} \text{DIST}(R_l^S, R_l^T)$$
$$+ \lambda_2 \sum_{i=1}^{n^{T,U}} \sum_{y_j \in \mathcal{Y}} S\left(P(y_j|f(\mathbf{x}_i^{T,U}))\right),$$

where l_{strt} and l_{end} denote the boundary indexes of the fully connected layers for adapting the distributions.

There are some other impressive works. For example, Long *et al.* constructed residual transfer networks for domain adaptation, which is motivated by deep residual learning [146]. Besides, another work by Long *et al.* proposes the Joint Adaptation Network (JAN) [147], which adapts the joint distribution difference of multiple layers. Sun and Saenko extended CORAL for deep domain adaptation and proposed an approach termed Deep CORAL (DCORAL), in which the CORAL loss is added to minimize the feature covariance [148]. Chen *et al.* realized that the instances with the same label should be close to each other in the feature space, and they not only add the CORAL loss but also add an instance-based class-level discrepancy loss [149]. Pan *et al.* constructed three prototypical networks (corresponding to $\mathcal{D}_S, \mathcal{D}_T$ and $\mathcal{D}_S \cup \mathcal{D}_T$) and incorporated the thought of multi-model consensus. They also adopt pseudo-label strategy and adapt both the instance-level and class-level discrepancy [150]. Kang *et al.* proposed the Contrastive Adaptation Network (CAN), which is based on the discrepancy metric termed contrastive domain discrepancy [151]. Zhu *et al.* aimed to adapt the extracted multiple feature representations and proposed the Multi-Representation Adaptation Network (MRAN) [152].

Deep learning technique can also be used for multi-source transfer learning. For example, the work by Zhu *et al.* proposes a framework that is referred to as Multiple Feature Spaces Adaptation Network (MFSAN) [153]. The architecture of MFSAN consists of a common-feature extractor, m^S domain-specific feature extractors, and m^S domain-specific

classifiers. The corresponding schematic diagram is shown below.

$$X_1^S \cdots X_k^S \cdots X_{m^S}^S \xrightarrow[\text{Extractor}]{\text{Common}} Q_1^S \cdots Q_k^S \cdots Q_{m^S}^S \xrightarrow[\text{Extractors}]{\text{Domain-Specific}}$$
$$X^T \qquad\qquad\qquad Q^T$$

$$R_1^S \cdots R_k^S \cdots R_{m^S}^S \xrightarrow[\text{Classifiers}]{\text{Domain-Specific}} \hat{Y}_1^S \cdots \hat{Y}_k^S \cdots \hat{Y}_{m^S}^S$$
$$R_1^T \cdots R_k^T \cdots R_{m^S}^T \qquad\qquad \hat{Y}_1^T \cdots \hat{Y}_k^T \cdots \hat{Y}_{m^S}^T$$

In each iteration, MFSAN has the following steps.

1. *Common Feature Extraction*: For each source domain (denoted by $\mathcal{D}_{S_k}$ with $k = 1, \cdots, m^S$), the source-domain instances (denoted by X_k^S) are separately input to the common-feature extractor to produce instances in a common latent feature space (denoted by Q_k^S). Similar operations are also performed on the target-domain instances (denoted by X^T), which produces Q^T.

2. *Specific Feature Extraction*: For each source domain, the extracted common features Q_k^S is fed to the k-th domain-specific feature extractor. Meanwhile, Q^T is fed to all the domain-specific feature extractors, which results in R_k^T with $k = 1, \cdots, m^S$.

3. *Data Classification*: The output of the k-th domain-specific feature extractor is input to the k-th classifier. In this way, m^S pairs of the classification results are predicted in the form of probability.

4. *Parameter Updating*: The parameters of the network are updated to optimize the objective function.

There are three objectives in MFSAN, i.e., classification error minimization, distribution adaptation, and consensus regularization. The objective function is given by:

$$\min_{\Theta} \sum_{i=1}^{m^S} \mathcal{L}(\hat{Y}_i^S, Y_i^S) + \lambda_1 \sum_{i=1}^{m^S} \text{MMD}(R_i^S, R_i^T)$$
$$+ \lambda_2 \sum_{i \neq j}^{m^S} \left| \hat{Y}_i^T - \hat{Y}_j^T \right|,$$

where the first term represents the classification error of the labeled source-domain instances, the second term measures the distribution difference, and the third term measures the discrepancy of the predictions on the target-domain instances.

5.4.2 Adversarial Deep Learning

The thought of adversarial learning can be integrated into deep-learning-based transfer learning approaches. As mentioned above, in the DAN framework, the network Θ and the kernel κ play a minimax game, which reflects the thought of adversarial learning. However, the DAN framework is a little different from the traditional GAN-based methods in terms of the adversarial matching. In the DAN framework, there is only a few parameters to be optimized in the max game, which makes the optimization easier to achieve equilibrium. Before introducing the adversarial transfer learning approaches, let us briefly review the original GAN framework and the related work.

The original GAN [139], which is inspired by the two-player game, is composed of two models, a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. The generator produces the counterfeits of the true data for the purpose of confusing the discriminator and making the discriminator produce wrong detection.

The discriminator is fed with the mixture of the true data and the counterfeits, and it aims to detect whether a data is the true one or the fake one. These two models actually play a two-player minimax game, and the objective function is as follows:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} \mathbb{E}_{\mathbf{x} \sim P_{\text{true}}} [\log \mathcal{D}(\mathbf{x})] + \mathbb{E}_{\tilde{\mathbf{z}} \sim P_{\tilde{\mathbf{z}}}} [\log (1 - \mathcal{D}(\mathcal{G}(\tilde{\mathbf{z}})))],$$

where $\tilde{\mathbf{z}}$ represents the noise instances (sampled from a certain noise distribution) used as the input of the generator for producing the counterfeits. The entire GAN can be trained by using the back-propagation algorithm. When the two-player game achieves equilibrium, the generator can produce almost true-looking instances.

Motivated by GAN, many transfer learning approaches are established based on the assumption that a good feature representation contains almost no discriminative information about the instances' original domains. For example, the work by Ganin *et al.* proposes a deep architecture termed Domain-Adversarial Neural Network (DANN) for domain adaptation [154], [155]. DANN assumes that there is no labeled target-domain instance to work with. Its architecture consists of a feature extractor, a label predictor, and a domain classifier. The corresponding diagram is as follows.

$$\xrightarrow[\text{Predictor}]{\text{Label}} \begin{matrix} \hat{Y}^{S,L} \\ \hat{Y}^{T,U} \end{matrix}$$
$$\begin{matrix} X^{S,L} \\ X^{T,U} \end{matrix} \xrightarrow[\text{Extractor}]{\text{Feature}} \left. \begin{matrix} Q^{S,L} \\ Q^{T,U} \end{matrix} \right\} \xrightarrow[\text{Classifier}]{\text{Domain}} \begin{matrix} \hat{S} \\ \hat{T} \end{matrix} \left(\text{Domain Label}\right)$$

The feature extractor acts like the generator, which aims to produce the domain-independent feature representation for confusing the domain classifier. The domain classifier plays the role like the discriminator, which attempts to detect whether the extracted features come from the source domain or the target domain. Besides, the label predictor produces the label prediction of the instances, which is trained on the extracted features of the labeled source-domain instances, i.e., $Q^{S,L}$. DANN can be trained by inserting a special gradient reversal layer (GRL). After the training of the whole system, the feature extractor learns the deep feature of the instances, and the output $\hat{Y}^{T,U}$ is the predicted labels of the unlabeled target-domain instances.

There are some other related impressive works. The work by Tzeng *et al.* proposes a unified adversarial domain adaptation framework [156]. The work by Shen *et al.* adopts Wasserstein distance for domain adaptation [59]. Hoffman *et al.* adopted cycle-consistency loss to ensure the structural and semantic consistency [157]. Long *et al.* proposed the Conditional Domain Adversarial Network (CDAN), which utilizes a conditional domain discriminator to assist adversarial adaptation [158]. Zhang *et al.* adopted a symmetric design for the source and the target classifiers [159]. Zhao *et al.* utilized domain adversarial networks to solve the multi-source transfer learning problem [160]. Yu *et al.* proposed a dynamic adversarial adaptation network [161].

Some approaches are designed for some special scenarios. Take the partial transfer learning as an example. The partial transfer learning approaches are designed for the scenario that the target-domain classes are less than the source-domain classes, i.e., $\mathcal{Y}^S \subseteq \mathcal{Y}^T$. In this case, the source-domain instances with different labels may have different

importance for domain adaptation. To be more specific, the source-domain and the target-domain instances with the same label are more likely to be potentially associated. However, since the target-domain instances are unlabeled, how to identify and partially transfer the important information from the labeled source-domain instances is a critical issue.

The paper by Zhang *et al.* proposes an approach for partial domain adaptation, which is called Importance Weighted Adversarial Nets-Based Domain Adaptation (IWANDA) [162]. The architecture of IWANDA is different from that of DANN. DANN adopts one common feature extractor based on the assumption that there exists a common feature space where $Q^{S,L}$ and $Q^{T,U}$ have the similar distribution. However, IWANDA uses two domain-specific feature extractors for the source and the target domains, respectively. Specifically, IWANDA consists of two feature extractors, two domain classifiers, and one label predictor. The diagram of IWANDA is presented below.

$$\xrightarrow[\text{Predictor}]{\text{Label}} \begin{matrix} \hat{Y}^{S,L} \\ \hat{Y}^{T,U} \end{matrix}$$

$$X^{S,L} \xrightarrow[\text{Extractor}]{\text{Source Feature}} \begin{matrix} Q^{S,L} \\ Q^{T,U} \end{matrix} \left.\begin{matrix} \\ \end{matrix}\right\} \xrightarrow[+\hat{Y}^{T,U}]{+\boldsymbol{\beta}^S} \xrightarrow[\text{Classifier}]{\text{2nd Domain}} \begin{matrix} \hat{S}_2 \\ \hat{T}_2 \end{matrix}$$

$$X^{T,U} \xrightarrow[\text{Extractor}]{\text{Target Feature}}$$

$$\xrightarrow[\text{Classifier}]{\text{1st Domain}} \begin{matrix} \hat{S}_1 \\ \hat{T}_1 \end{matrix} \xrightarrow[\text{Function}]{\text{Weight}} \boldsymbol{\beta}^S$$

Before training, the source feature extractor and the label predictor are pre-trained on the labeled source-domain instances. These two components are frozen in the training process, which means that only the target feature extractor and the domain classifiers should be optimized. In each iteration, the above network is optimized by taking the following steps.

1. *Instance Weighting*: In order to solve the partial transfer issue, the source-domain instances are assigned with weights based on the output of the first domain classifier. The first domain classifier is fed with $Q^{S,L}$ and $Q^{T,U}$, and then outputs the probabilistic predictions of their domains. If a source domain instance is predicted with a high probability of belonging to the target domain, this instance is highly likely to associate with the target domain. Thus, this instance is assigned with a larger weight and vice versa.

2. *Prediction Making*: The label predictor outputs the label predictions of the instances. The second classifier predicts which domain an instance belongs to.

3. *Parameter Updating*: The first classifier is optimized to minimize the domain classification error. The second classifier plays a minmax game with the target feature extractor. This classifier aims to detect whether a instance is the instance from the target domain or the weighted instance from the source domain, and to reduce the uncertainty of the label prediction $\hat{Y}^{T,U}$. The target feature extractor aims to confuse the second classifier. These components can be optimized in a similar way to GAN or by inserting a GRL.

In addition to IWANDA, the work by Cao *et al.* constructs the selective adversarial network for partial transfer learning [163]. There are some other studies related to

transfer learning. For example, the work by Wang *et al.* proposes a minimax-based approach to select high-quality source-domain data [164]. Chen *et al.* investigated the transferability and the discriminability in the adversarial domain adaptation, and proposed a spectral penalization approach to boost the existing adversarial transfer learning methods [165].

6 APPLICATION

In previous sections, a number of representative transfer learning approaches are introduced, which have been applied to solving a variety of text-related/image-related problems in their original papers. For example, MTrick [122] and TriTL [123] utilize the matrix factorization technique to solve cross-domain text classification problems; the deep-learning-based approaches such as DAN [137], DCORAL [148], and DANN [154], [155] are applied to solving image classification problems. Instead of focusing on the general text-related or image-related applications, in this section, we mainly focus on the transfer learning applications in specific areas such as medicine, bioinformatics, transportation, and recommender systems.

6.1 Medical Application

Medical imaging plays an important role in the medical area, which is a powerful tool for diagnosis. With the development of computer technology such as machine learning, computer-aided diagnosis has become a popular and promising direction. Note that medical images are generated by special medical equipment, and their labeling often relies on experienced doctors. Therefore, in many cases, it is expensive and hard to collect sufficient training data. Transfer learning technology can be utilized for medical imaging analysis. A commonly used transfer learning approach is to pre-train a neural network on the source domain (e.g., ImageNet, which is an image database containing more than fourteen million annotated images with more than twenty thousand categories [166]) and then finetune it based on the instances from the target domain.

For example, Maqsood *et al.* finetuned the AlexNet [138] for the detection of Alzheimer's disease [167]. Their approach has the following four steps. First, the MRI images from the target domain are pre-processed by performing contrast stretching operations. Second, the AlexNet architecture [138] is pre-trained over ImageNet [166] (i.e., the source domain) as a starting point to learn the new task. Third, the convolutional layers of AlexNet are fixed, and the last three fully connected layers are replaced by the new ones including one softmax layer, one fully connected layer, and one output layer. Finally, the modified AlexNet is finetuned by training on the Alzheimer's dataset [168] (i.e., the target domain). The experimental results show that the proposed approach achieves the highest accuracy for the multi-class classification problem (i.e, Alzheimer's stage detection).

Similarly, Shin *et al.* finetuned the pre-trained deep neural network for solving the computer-aided detection problems [169]. Byra *et al.* utilized the transfer learning technology to help assess knee osteoarthritis [170]. In addition to imaging analysis, transfer learning has some other applications in the medical area. For example, the work by Tang *et*

al. combines the active learning and the domain adaptation technologies for the classification of various medical data [171]. Zeng *et al.* utilized transfer learning for automatically encoding ICD-9 codes that are used to describe a patient's diagnosis [172].

6.2 Bioinformatics Application

Biological sequence analysis is an important task in the bioinformatics area. Since the understanding of some organisms can be transferred to other organisms, transfer learning can be applied to facilitate the biological sequence analysis. The distribution difference problem exists significantly in this application. For example, the function of some biological substances may remain unchanged but with the composition changed between two organisms, which may result in the marginal distribution difference. Besides, if two organisms have a common ancestor but with long evolutionary distance, the conditional distribution difference would be significant. The work by Schweikert *et al.* uses the mRNA splice site prediction problem as the example to analyze the effectiveness of transfer learning approaches [173]. In their experiments, the source domain contains the sequence instances from a well-studied model organism, i.e., *C. elegans*, and the target organisms include two additional nematodes (i.e., *C. remanei* and *P. pacificus*), *D. melanogaster*, and the plant *A. thaliana*. A number of transfer learning approaches, e.g., FAM [64] and the variant of KMM [5], are compared with each other. The experimental results show that transfer learning can help improve the classification performance.

Another widely encountered task in the bioinformatics area is gene expression analysis, e.g., predicting associations between genes and phenotypes. In this application, one of the main challenges is the data sparsity problem, since there is usually very little data of the known associations. Transfer learning can be used to leverage this problem by providing additional information and knowledge. For example, Petegrosso *et al.* [174] proposed a transfer learning approach to analyze and predict the gene-phenotype associations based on the Label Propagation Algorithm (LPA) [175]. LPA utilizes the Protein-Protein Interaction (PPI) network and the initial labeling to predict the target associations based on the assumption that the genes that are connected in the PPI network should have the similar labels. The authors extended LPA by incorporating multi-task and transfer-learning technologies. First, Human Phenotype Ontology (HPO), which provides a standardized vocabulary of phenotypic features of human diseases, is utilized to form the auxiliary task. In this way, the associations can be predicted by utilizing phenotype paths and both the linkage knowledge in HPO and in the PPI network; the interacted genes in PPI are more likely to be associated with the same phenotype and the connected phenotypes in HPO are more likely to be associated with the same gene. Second, Gene Ontology (GO), which contains the association information between gene functions and genes, is used as the source domain. Additional regularizers are designed, and the PPI network and the common genes are used as the bridge for knowledge transfer. The gene-GO term and gene-HPO phenotype associations are constructed simultaneously for all the genes in the PPI network. By transferring additional knowledge, the predicted gene-phenotype associations can be more reliable.

Transfer learning can also be applied to solving the PPI prediction problems. Xu *et al.* [176] proposed an approach to transfer the linkage knowledge from the source PPI network to the target one. The proposed approach is based on the collective matrix factorization technique [177], where a factor matrix is shared across domains.

6.3 Transportation Application

One application of transfer learning in the transportation area is to understand the traffic scene images. In this application, a challenge problem is that the images taken from a certain location often suffer from variations because of different weather and light conditions. In order to solve this problem, Di *et al.* proposed an approach that attempts to transfer the information of the images that were taken from the same location in different conditions [178]. In the first step, a pre-trained network is finetuned to extract the feature representations of images. In the second step, the feature transformation strategy is adopted to construct a new feature representation. Specifically, the dimension reduction algorithm (i.e., partial least squares regression [179]) is performed on the extracted features to generate low-dimension features. Then, a transformation matrix is learned to minimize the domain discrepancy of the dimension-reduced data. Next, the subspace alignment operations are adopted to further reduce the domain discrepancy. Note that, although images under different conditions often have different appearances, they often have the similar layout structure. Therefore, in the final step, the cross-domain dense correspondences are established between the test image and the retrieved best matching image at first, and then the annotations of the best matching image are transferred to the test image via the Markov random field model [180], [181].

Transfer learning can also be applied to the task of driver behavior modeling. In this task, sufficient personalized data of each individual driver are usually unavailable. In such situations, transferring the knowledge contained in the historical data for the newly-involved driver is a promising alternative. For example, Lu *et al.* proposed an approach to driver model adaptation in lane-changing scenarios [182]. The source domain contains the sufficient data describing the behavior of the source drivers, while the target domain has a few numbers of data about the target driver. In the first step, the data from both domains are pre-processed by performing PCA to generate low-dimension features. The authors assume that the source and the target data are from two manifolds. Therefore, in the second step, a manifold alignment approach is adopted for domain adaptation. Specifically, the dynamic time warping algorithm [183] is applied to measuring similarity and finding the corresponding source-domain data point of each target-domain data point. Then, local Procrustes analysis [184] is adopted to align the two manifolds based on the obtained correspondences between data points. In this way, the data from the source domain can be transferred to the target domain. And in the final step, a stochastic modeling method (e.g., Gaussian mixture regression [185]) is used to model

the behavior of the target driver. The experimental results demonstrate that the transfer learning approach can help the target driver even when few target-domain data are available. Besides, the results also show that when the number of target instances are very small or very large, the superiority of their approach is not obvious. This may because the relationship across domains cannot be found exactly with few target-domain instances, and in the case of sufficient target-domain instances, the necessity of transfer learning is reduced.

Besides, there are some other applications of transfer learning in the transportation area. For example, Liu *et al.* applied transfer learning to driver poses recognition [186]. Wang *et al.* adopted the regularization technique in transfer learning for vehicle type recognition [187]. Transfer learning can also be utilized for anomalous activity detection [188], [189], traffic sign recognition [190], etc.

6.4 Recommender-System Application

Due to the rapid increase of the amount of information, how to effectively recommend the personalized content for individual users is an important issue. In the field of recommender systems, some traditional recommendation methods, e.g., factorization-based collaborative filtering, often rely on the factorization of the user-item interaction matrix to obtain the predictive function. These methods often require a large amount of training data to make accurate recommendations. However, the necessary training data, e.g., the historical interaction data, are often sparse in real-world scenarios. Besides, for new registered users or new items, traditional methods are often hard to make effective recommendations, which is also known as the cold-start problem.

Recognizing the above-mentioned problems in recommender systems, kinds of transfer learning approaches, e.g., instance-based and feature-based approaches, have been proposed. These approaches attempt to make use of the data from other recommender systems (i.e., the source domains) to help construct the recommender system in the target domain. Instance-based approaches mainly focus on transferring different types of instances, e.g., ratings, feedbacks, and examinations, from the source domain to the target domain. The work by Pan *et al.* [191] leverages the uncertain ratings (represented as rating distributions) of the source domain for knowledge transfer. Specifically, the source-domain uncertain ratings are used as constraints to help complete the rating matrix factorization task on the target domain. Hu *et al.* [192] proposed an approach termed transfer meeting hybrid, which extracts the knowledge from unstructured text by using an attentive memory network and selectively transfer the useful information.

Feature-based approaches often leverage and transfer the information from a latent feature space. For example, Pan *et al.* proposed an approach termed Coordinate System Transfer (CST) [193] to leverage both the user-side and the item-side latent features. The source-domain instances come from another recommender system, sharing common users and items with the target domain. CST is developed based on the assumption that the principle coordinates, which reflect the tastes of users or the factors of items, characterize the domain-independent structure and are transferable across domains. CST first constructs two principle coordinate systems, which are actually the latent features of users and items, by applying sparse matrix tri-factorization on the source-domain data, and then transfer the coordinate systems to the target domain by setting them as constraints. The experimental results show that CST significantly outperforms the non-transfer baselines (i.e., average filling model and latent factorization model) in all data sparsity levels [193].

There are some other studies about cross-domain recommendation [194], [195], [196], [197]. For example, He *et al.* proposed a transfer learning framework based on the Bayesian neural network [198]. Zhu *et al.* [199] proposed a deep framework, which first generates the user and item feature representations based on the matrix factorization technique, and then employs a deep neural network to learn the mapping of features across domains. Yuan *et al.* [200] proposed a deep domain adaptation approach based on autoencoders and a modified DANN [154], [155] to extract and transfer the instances from rating matrices.

6.5 Other Applications

Communication Application: In addition to WiFi localization tasks [2], [36], transfer learning has also been employed in wireless-network applications. For example, Bastug *et al.* proposed a caching mechanism [201]; the knowledge contained in contextual information, which is extracted from the interactions between devices, is transferred to the target domain. Besides, some studies focus on the energy saving problems. The work by Li *et al.* proposes an energy saving scheme for cellular radio access networks, which utilizes the transfer-learning expertise [202]. The work by Zhao and Grace applies transfer learning to topology management for reducing energy consumption [203].

Urban-Computing Application: With a large amount of data related to our cities, urban-computing is a promising researching track in directions of traffic monitoring, health care, social security, etc. Transfer learning has been applied to alleviate the data scarcity problem in many urban computing applications. For example, Guo *et al.* [204] proposed an approach for chain store site recommendation, which leverages the knowledge from semantically-relevant domains (e.g., other cities with the same store and other chain stores in the target city) to the target city. Wei *et al.* [205] proposed a flexible multi-modal transfer learning approach that transfers knowledge from a city that have sufficient multi-model data and labels to the target city to alleviate the data sparsity problem.

Transfer learning has been applied to some recognition tasks such as hand gesture recognition [206], face recognition [207], activity recognition [208], and speech emotion recognition [209]. Besides, transfer-learning expertise has also been incorporated into some other areas such as sentiment analysis [28], [96], [210], fraud detection [211], social network [212], and hyperspectral image analysis [54], [213].

7 EXPERIMENT

Transfer learning techniques have been successfully applied in many real-world applications. In this section, we perform

experiments to evaluate the performance of some representative transfer learning models[1] [214] of different categories on two mainstream research areas, i.e., object recognition and text classification. The datasets are introduced at first. Then, the experimental results and further analyses are provided.

7.1 Dataset and Preprocessing

Three datasets are studied in the experiments, i.e., Office-31, Reuters-21578, and Amazon Reviews. For simplicity, we focus on the classification tasks. The statistical information of the preprocessed datasets is listed in Table 3.

- **Amazon Reviews**[2] [107] is a multi-domain sentiment dataset which contains product reviews taken from Amazon.com of four domains (Books, Kitchen, Electronics and DVDs). Each review in the four domains has a text and a rating from zero to five. In the experiments, the ratios that are less than three are defined as the negative ones, while others are defined as the positive ones. The frequency of each word in all reviews is calculated. Then, the five thousand words with the highest frequency are selected as the attributes of each review. In this way, we finally have 1000 positive instances, 1000 negative instances, and about 5000 unlabeled instances in each domain. In the experiments, every two of the four domains are selected to generate twelve tasks.

- **Reuters-21578**[3] is a dataset for text categorization, which has a hierarchical structure. The dataset contains 5 top categories (Exchanges, Orgs, People, Places, Topics). In out experiment, we use the top three big category Orgs, People and Places to generate three classification tasks (Orgs vs People, Orgs vs Places and People vs Places). In each task, the subcategories in the corresponding two categories are separately divided into two parts. Then, the resultant four parts are used as the components to form two domains. Each domain has about 1000 instances, and each instance has about 4500 features. Specifically, taking the task Orgs vs People as an example, one part from Orgs and one part from People and combined to form the source domain; similarly, the rest two parts form the target domain. Note that the instances in the three categories are all labeled. In order to generate the unlabeled instances, the labeled instances are selected from the dataset, and their labels are ignored.

- **Office-31** [215] is an object recognition dataset which contains thirty-one categories and three domains, i.e., Amazon, Webcam, and DSLR. These three domains have 2817, 498, and 795 instances, respectively. The images in Amazon are the online e-commerce pictures taken from Amazon.com. The images in Webcam are the low-resolution pictures taken by web cameras. And the images in DSLR are the high-resolution pictures taken by DSLR cameras. In the experiments, every two of the three domains (with the order considered) are selected as the source and the target domains, which results in six tasks.

1. https://github.com/FuzhenZhuang/Transfer-Learning-Toolkit
2. http://www.cs.jhu.edu/ mdredze/datasets/sentiment/
3. https://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection

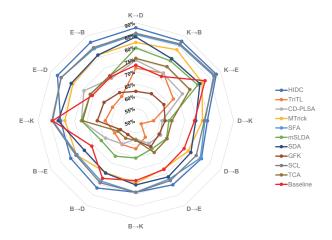Fig. 5. Comparison results on Amazon Review

Model	K→D	K→B	K→E	D→K	D→B	D→E	B→K	B→D	B→E	E→K	E→D	E→B	
HIDC							0.7825	0.8175	0.8075	0.8075	0.87	0.87	0.8
TriTL	0.715	0.725	0.6775	0.5725	0.525	0.5775	0.615	0.6125	0.6	0.625	0.61	0.615	0.6
CD-PLSA	0.7475	0.7225	0.72	0.6075	0.6175	0.6075	0.575	0.61	0.6425	0.7225	0.745	0.7	0.6
MTrick	0.82	0.835	0.8125	0.7725	0.7475	0.7275	0.755	0.745	0.78	0.79	0.7975	0.81	0.7
SFA	0.8525	0.8575	0.8675	0.7825	0.805	0.775	0.7925	0.785	0.84	0.8525	0.84		0.8
mSLDA	0.7975	0.7825	0.7925	0.635	0.645	0.6325	0.6525	0.6675	0.6625	0.7225	0.715	0.7125	0.7
SDA					0.76	0.765	0.7625	0.7475	0.7425	0.8175	0.805	0.81	0.7
GFK	0.62	0.6275	0.6325	0.62	0.61	0.6225	0.58	0.565	0.5725	0.6575	0.65	0.6325	0.6
SCL	0.8575	0.8625	0.8725	0.78	0.785	0.7825	0.7925	0.7925	0.7825	0.8425	0.8525	0.845	0.8
TCA										0.73	0.715	0.7125	0.6
Baseline	0.727	0.709	0.827	0.74	0.728	0.73	0.745	0.772	0.708	0.84	0.706	0.707	0.7

7.2 Experiment Setting

Experiments are conducted to compare some representative transfer learning models. Specifically, eight algorithms are performed on the dataset Office-31 for solving the object recognition problem. Besides, fourteen algorithms are performed and evaluated on the dataset Reuters-21578 for solving the text classification problem. In the sentiment classification problem, eleven algorithms are performed on Amazon Reviews. The classification results are evaluated by accuracy, which is defined as follows:

$$\text{accuracy} = \frac{|\{\mathbf{x}|\mathbf{x}_i \in \mathcal{D}_{\text{test}} \wedge f(\mathbf{x}_i) = y_i\}|}{|\mathcal{D}_{\text{test}}|}$$

where $\mathcal{D}_{\text{test}}$ denotes the test data and y denotes the truth classification label; $f(\mathbf{x})$ represents the predicted classification result. Note that some algorithms need the base classifier. In these cases, an SVM with a linear kernel is adopted as the base classifier in the experiments. Besides, the source-domain instances are all labeled. And for the performed algorithms (except TrAdaBoost), the target-domain instances are unlabeled. Each algorithm was executed three times, and the average results are adopted as our experimental results.

The evaluated transfer learning models include: HIDC [93], TriTL [123], CD-PLSA [91], [92], MTrick [122], SFA [106], mSLDA [98], [99], SDA [96], GFK [102], SCL [94], TCA [36], [78], CoCC [41], JDA [38], TrAdaBoost [31], DAN [137], DCORAL [148], MRAN [152], CDAN [158], DANN [154], [155], JAN [147], and CAN [151].

7.3 Experiment Result

In this subsection, we compare over twenty algorithms on three datasets in total. The parameters of all algorithms are set to the default values or the recommended values mentioned in the original papers. The experimental results are presented in Tables 4, 5, and 6 corresponding to Amazon Reviews, Reuters-21578, and Office-31, respectively. In order to allow readers to understand the experimental results more intuitively, three radar maps, i.e., Figs. 5, 6, and 7, are provided, which visualize the experimental results. In the radar maps, each direction represents a task. The general performance of an algorithm is demonstrated by a polygon

TABLE 3
Statistical information of the preprocessed datasets.

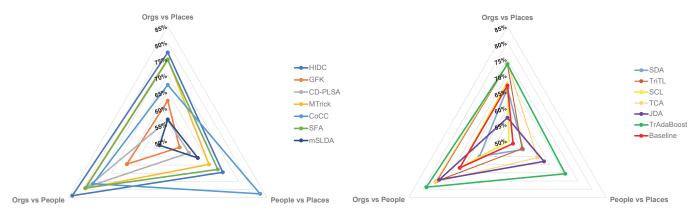| Area	Dataset	Domain	Attribute	Total Instances	Tasks
Sentiment Classification	Amazon Reviews	4	5000	27677	12
Text Classification	Reuters-21578	3	4772	6570	3
Object Recognition	Office-31	3	800	4110	6

Fig. 6. Comparison results on Reuters-21578.

TABLE 4
Accuracy performance on the Amazon Reviews of four domains: Kitchen (K), Electronics (E), DVDs (D) and Books (B).

Model	K→D	K→B	K→E	D→K	D→B	D→E	B→K	B→D	B→E	E→K	E→D	E→B	Average
HIDC	0.8750	0.8601	0.7925	0.8100	0.8025	0.7758	0.8175	0.8075	0.8075	0.8075	0.9800	0.8700	0.8338
TriTL	0.7250	0.7844	0.5725	0.5250	0.5775	0.7388	0.6125	0.6860	0.6250	0.6950	0.7420	0.6150	0.6225
CD-PLSA	0.7225	0.7280	0.6075	0.6175	0.6075	0.5750	0.6100	0.6425	0.7225	0.7450	0.7420	0.7000	0.6681
MTrick	0.8200	0.8350	0.8125	0.7725	0.7475	0.7275	0.7550	0.7450	0.7800	0.7900	0.7975	0.8100	0.7827
SFA	0.8525	0.8575	0.8675	0.7825	0.8050	0.7750	0.7925	0.7850	0.7775	0.8400	0.8525	0.8400	0.8190
mSLDA	0.7975	0.7825	0.7925	0.6350	0.6450	0.6325	0.6625	0.6675	0.7225	0.7150	0.7150	0.7125	0.7015
SDA	0.8425	0.7925	0.8025	0.7450	0.7600	0.7650	0.7625	0.7475	0.7425	0.8175	0.8050	0.8100	0.7827
GFK	0.6200	0.6275	0.6325	0.6200	0.6100	0.6225	0.5800	0.5650	0.5725	0.6575	0.6500	0.6325	0.6158
SCL	0.8575	0.8625	0.8725	0.7800	0.7850	0.7825	0.7925	0.7925	0.7825	0.8425	0.8525	0.8450	0.8206
TCA	0.7550	0.7550	0.7550	0.6475	0.6475	0.6500	0.5800	0.5825	0.5850	0.7175	0.7150	0.7125	0.6752
Baseline	0.7270	0.7090	0.8270	0.7400	0.7280	0.7300	0.7450	0.7720	0.7080	0.8400	0.7060	0.7070	0.7449

whose vertices representing the accuracy of the algorithm for dealing with different tasks.

Table 4 shows the experimental results on Amazon Reviews. The baseline is a linear classifier trained only on the source domain (here we directly use the results from the paper [107]). Fig. 5 visualizes the results. As shown in Fig. 5, most algorithms are relatively well-performed when the source domain is electronics or kitchen, which indicates that these two domains may contains more transferable information than the other two domains. In addition, it can be observed that HIDC, SCL, SFA, MTrick and SDA perform well and relatively stable in all the twelve tasks. Meanwhile, other algorithms, especially mSLDA, CD-PLSA, and TriTL, are relatively unstable; the performance of them fluctuates in a range about twenty percent. TriTL has a relatively high accuracy on the tasks where the source domain is kitchen, but has a relatively low accuracy on other tasks.

The algorithms TCA, mSLDA, and CD-PLSA have similar performance on all the tasks with an accuracy about seventy percent on average. Among the well-performed algorithms, HIDC and MTrick are based on feature reduction (feature clustering), while the others are based on feature encoding (SDA), feature alignment (SFA), and feature selection (SCL). Those strategies are currently the mainstreams of feature-based transfer learning.

Table 5 presents the comparison results on Reuter-21578 (here we directly use the results of the baseline and CoCC from papers [78] and [41]). The baseline is a regularized least square regression model trained only on the labeled target domain instances [78]. Fig. 6, which has the same structure of Fig. 5, visualizes the performance. For clarity, thirteen algorithms are divided into two parts that correspond to the two subfigures in Fig. 6. It can be observed that most algorithms are relatively well-performed for Orgs vs Places
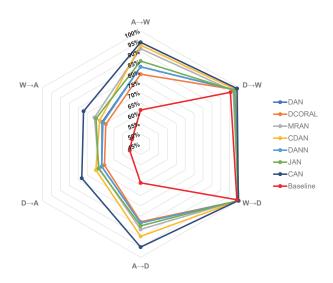
nomenon indicates that the discrepancy between People and Places may be relatively large. TrAdaBoost has a relatively good performance in this experiment because it uses the labels of the instances in the target domain to reduce the impact of the distribution difference. Besides, the algorithms HIDC, SFA, and MTrick have relatively consistent performance in the three tasks. These algorithms are also well-performed in the previous experiment on Amazon Reviews. In addition, the top two well-performed algorithms in terms of People vs Places are CoCC and TrAdaBoost.

In the third experiment, seven deep-learning-based transfer learning models (i.e., DAN, DCORAL, MRAN, CDAN, DANN, JAN, and CAN) and the baseline (i.e., the Alexnet [138], [140] pre-trained on ImageNet [166] and then directly trained on the target domain) are performed on the dataset Office-31 (here we directly use the results of CDAN, JAN, CAN, and the baseline from the original papers [137], [147], [151], [158]). The ResNet-50 [144] is used as the backbone network for all these three models. The experimental results are provided in Table 6 and the average performance is visualized in Fig. 7. As shown in Fig. 7, all of these seven algorithms have excellent performance, especially on the tasks D → W and W → D, whose accuracy is very close to one hundred percent. This phenomenon reflects the superiority of the deep-learning based approaches, and is consistent with the fact that the difference between Webcam and DSLR is smaller than that between Webcam/DSLR and Amazon. Clearly, CAN outperforms the other six algorithms. In all the six tasks, the performance of DANN is similar to that of DAN, and is better than that of DCORAL, which indicates the effectiveness and the practicability of incorporating adversarial learning.

It is worth mentioning that, in the above experiments, the performance of some algorithms is not ideal. One reason is that we use the default parameter settings provided in the algorithms' original papers, which may not be suitable for the dataset we selected. For example, GFK was originally designed for object recognition, and we directly adopt it into text classification in the first experiment, which turns out to produce an unsatisfactory result (having about sixty-two percent accuracy on average). The above experimental results are just for reference. These results demonstrate that some algorithms may not be suitable for the datasets of certain domains. Therefore, it is important to choose the appropriate algorithms as the baselines in the process of research. Besides, in practical applications, it is also necessary to find a suitable algorithm.

8 CONCLUSION AND FUTURE DIRECTION

In this survey paper, we have summarized the mechanisms and the strategies of transfer learning from the perspectives of data and model. The survey gives the clear definitions about transfer learning and manages to use a unified symbol system to describe a large number of representative transfer learning approaches and related works. We have basically introduced the objectives and strategies in transfer learning based on data-based interpretation and model-based interpretation. Data-based interpretation introduces the objectives, the strategies, and some transfer learning

Fig. 7. Comparison results on Office-31

TABLE 5
Accuracy performance on the Reuters-21578 of three domains: Orgs, People, and Places.

Model	Orgs vs Places	People vs Places	Orgs vs People	Average
HIDC	0.7698	0.6945	0.8375	0.7673
TriTL	0.7338	0.5517	0.7505	0.6787
CD-PLSA	0.5624	0.5749	0.7826	0.6400
MTrick	0.7494	0.6457	0.7930	0.7294
CoCC	0.6704	0.8264	0.7644	0.7537
SFA	0.7468	0.6768	0.7906	0.7381
mSLDA	0.5645	0.6064	0.5289	0.5666
SDA	0.6603	0.5556	0.5992	0.6050
GFK	0.6220	0.5417	0.6446	0.6028
SCL	0.6794	0.5046	0.6694	0.6178
TCA	0.7368	0.6065	0.7562	0.6998
JDA	0.5694	0.6296	0.7424	0.6471
TrAdaBoost	0.7336	0.7052	0.7879	0.7422
Baseline	0.6683	0.5198	0.6696	0.6192

TABLE 6
Accuracy performance on Office-31 of three domains: Amazon (A), Webcam (W), and DSLR (D).

Model	A → W	D→W	W→D	A→D	D → A	W→A	Average
DAN	0.826	0.977	1.00	0.831	0.668	0.666	0.828
DCORAL	0.790	0.980	1.00	0.827	0.653	0.645	0.816
MRAN	0.914	0.969	0.998	0.864	0.683	0.709	0.856
CDAN	0.931	0.982	1.00	0.898	0.701	0.680	0.865
DANN	0.826	0.978	1.00	0.833	0.668	0.661	0.828
JAN	0.854	0.974	0.998	0.847	0.686	0.700	0.843
CAN	0.945	0.991	0.998	0.950	0.780	0.770	0.906
Baseline	0.616	0.954	0.990	0.638	0.511	0.498	0.701

and Orgs vs People, but poor for People vs Places. This phe-

approaches from the data perspective. Similarly, model-based interpretation introduces the mechanisms and the strategies of transfer learning but from the model level. The applications of transfer learning have also been introduced. At last, experiments have been conducted to evaluate the performance of representative transfer learning models on two mainstream area, i.e., object recognition and text categorization. The comparisons of the models have also been given, which reflects that the selection of the transfer learning model is an important research topic as well as a complex issue in practical applications.

Several directions are available for future research in the transfer learning area. First, transfer learning techniques can be further explored and applied to a wider range of applications. And new approaches are needed to solve the knowledge transfer problems in more complex scenarios. For example, in real-world scenarios, sometimes the user-relevant source-domain data comes from another company. In this case, how to transfer the knowledge contained in the source domain while protecting user privacy is an important issue. Second, how to measure the transferability across domains and avoid negative transfer is also an important issue. Although there have been some studies on negative transfer, negative transfer still needs further systematic analyses [3]. Third, the interpretability of transfer learning also needs to be investigated further [216]. Finally, theoretical studies can be further conducted to provide theoretical support for the effectiveness and applicability of transfer learning. As a popular and promising area in machine learning, transfer learning shows some advantages over traditional machine learning such as less data dependency and less label dependency. We hope our work can help readers have a better understanding of the research status and the research ideas.

ACKNOWLEDGMENTS

REFERENCES

[1] D.N. Perkins and G. Salomon, *Transfer of Learning*. Oxford, England: Pergamon, 1992.

[2] S.J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[3] Z. Wang, Z. Dai, B. Póczos, and J. Carbonell, "Characterizing and avoiding negative transfer," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, Jun. 2019, pp. 11293–11302.

[4] K. Weiss, T.M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *J. Big Data*, vol. 3, no. 1, Dec. 2016.

[5] J. Huang, A.J. Smola, A. Gretton, K.M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Proc. 20th Annual Conference on Neural Information Processing Systems*, Vancouver, Dec. 2006, pp. 601–608.

[6] M. Sugiyama, T. Suzuki, S. Nakajima, H. Kashima, P. Bnau, and M. Kawanabe, "Direct importance estimation for covariate shift adaptation," *Ann. Inst. Stat. Math.*, vol. 60, no. 4, pp. 699–746, Dec. 2008.

[7] O. Day and T.M. Khoshgoftaar, "A survey on heterogeneous transfer learning," *J. Big Data*, vol. 4, no. 1, Dec. 2017.

[8] M.E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, pp. 1633–1685, Sep. 2009.

[9] H.B. Ammar, E. Eaton, J.M. Luna, and P. Ruvolo, "Autonomous cross-domain knowledge transfer in lifelong policy gradient reinforcement learning," in *Proc. 24th International Joint Conference on Artificial Intelligence*, Buenos Aires, Jul. 2015, pp. 3345–3351.

[10] P. Zhao and S.C.H. Hoi, "OTL: A framework of online transfer learning," in *Proc. 27th International Conference on Machine Learning*, Haifa, Jun. 2010, pp. 1231–1238.

[11] O. Chapelle, B. Schlkopf, and A. Zien, *Semi-supervised Learning*. Cambridge: MIT Press, 2010.

[12] S. Sun, "A survey of multi-view machine learning," *Neural Comput. Appl.*, vol. 23, no. 7–8, pp. 2031–2038, Dec. 2013.

[13] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," 2013, arXiv:1304.5634v1.

[14] J. Zhao, X. Xie, X. Xu, and S. Sun, "Multi-view learning overview: Recent progress and new challenges," *Inf. Fusion*, vol. 38, pp. 43–54, Nov. 2017.

[15] D. Zhang, J. He, Y. Liu, L. Si, and R. Lawrence, "Multi-view transfer learning with a large margin approach," in *Proc. 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, Aug. 2011, pp. 1208–1216.

[16] P. Yang and W. Gao, "Multi-view discriminant transfer learning," in *Proc. 23rd International Joint Conference on Artificial Intelligence*, Beijing, Aug. 2013, pp. 1848–1854.

[17] K.D. Feuz and D.J. Cook, "Collegial activity learning between heterogeneous sensors," *Knowl. Inf. Syst.*, vol. 53, pp. 337–364, Mar. 2017.

[18] Y. Zhang and Q. Yang, "An overview of multi-task learning," *Natl. Sci. Rev.*, vol. 5, no. 1, pp. 30–43, Jan. 2018.

[19] W. Zhang, R. Li, T. Zeng, Q. Sun, S. Kumar, J. Ye, and S. Ji, "Deep model based transfer and multi-task learning for biological image analysis," in *Proc. 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, Aug. 2015, pp. 1475–1484.

[20] A. Liu, N. Xu, W. Nie, Y. Su, and Y. Zhang, "Multi-domain and multi-task learning for human action recognition," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 853–867, Feb. 2019.

[21] X. Peng, Z. Huang, X. Sun, and K. Saenko, "Domain agnostic learning with disentangled representations," in *Proc. 36th International Conference on Machine Learning*, Long Beach, Jun. 2019, pp. 5102–5112.

[22] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowledge-Based Syst.*, vol. 80, pp. 14–23, May 2015.

[23] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A Survey on deep transfer learning," in *Proc. 27th International Conference on Artificial Neural Networks*, Rhodes, Oct. 2018, pp. 270–279.

[24] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[25] D. Cook, K.D. Feuz, and N.C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowl. Inf. Syst.*, vol. 36, no. 3, pp. 537–556, Sep. 2013.

[26] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2015.

[27] W. Pan, "A survey of transfer learning for collaborative recommendation with auxiliary data," *Neurocomputing*, vol. 177, pp. 447–453, Feb. 2016.

[28] R. Liu, Y. Shi, C. Ji, and M. Jia, "A Survey of sentiment analysis based on transfer learning," *IEEE Access*, vol. 7, pp. 85401–85412, Jun. 2019.

[29] Q. Sun, R. Chattopadhyay, S. Panchanathan, and J. Ye, "A two-stage weighting framework for multi-source domain adaptation," in *Proc. 25th Annual Conference on Neural Information Processing Systems*, Granada, Dec. 2011, pp. 505–513.

[30] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.

[31] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. 24th International Conference on Machine Learning*, Corvalis, Jun. 2007, pp. 193–200.

[32] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[33] Y. Yao and G. Doretto, "Boosting for transfer learning with multiple sources," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, Jun. 2010, pp. 1855–1862.

[34] J. Jiang and C. Zhai, "Instance weighting for domain adaptation in NLP," in *Proc. 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Jun. 2007, pp. 264–271.

[35] K.M. Borgwardt, A. Gretton, M.J. Rasch, H.-P. Kriegel, B. Scholkopf, and A.J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. 49–57, Jul. 2006.

[36] S.J. Pan, I.W. Tsang, J.T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis,"*IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[37] M. Ghifary, W.B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *Proc. Pacific Rim International Conference on Artificial Intelligence*, Gold Coast, Dec. 2014, pp. 898–904.

[38] M. Long, J. Wang, G. Ding, J. Sun, and P.S. Yu, "Transfer feature learning with joint distribution adaptation,"in *Proc. IEEE International Conference on Computer Vision*, Sydney, Dec. 2013, pp. 2200–2207.

[39] M. Long, J. Wang, G. Ding, S.J. Pan, and P.S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076-1089, May 2014.

[40] S. Kullback and R.A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[41] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, Aug. 2007, pp. 210–219.

[42] W. Dai, Q. Yang, G. Xue, and Y. Yu, "Self-taught clustering," in *Proc. 25th International Conference of Machine Learning*, Helsinki, Jul. 2008, pp. 200–207.

[43] J. Davis and P. Domingos, "Deep transfer via second-order Markov logic," in *Proc. 26th International Conference on Machine Learning*, Montreal, Jun. 2009, pp. 217–224.

[44] F. Zhuang, X. Cheng, P. Luo, S.J. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Proc. 24th International Joint Conference on Artificial Intelligence*, Buenos Aires, Jul. 2015, pp. 4119–4125.

[45] I. Dagan, L. Lee, and F. Pereira, "Similarity-based methods for word sense disambiguation," in *Proc. 35th Annual Meeting of the Association of Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics (ACL/EACL)*, Madrid, Jul. 1997, pp. 56–63.

[46] B. Chen, W. Lam, I. Tsang, and T. Wong, "Location and scatter matching for dataset shift in text mining," in *Proc. 10th IEEE International Conference on Data Mining*, Sydney, Dec. 2010, pp. 773–778.

[47] S. Dey, S. Madikeri, and P. Motlicek, "Information theoretic clustering for unsupervised domain-adaptation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Shanghai, Mar. 2016, pp. 5580–5584.

[48] W.-H. Chen, P.-C. Cho, and Y.-L. Jiang, "Activity recognition using transfer learning," *Sens. Mater.*, vol. 29, no. 7, pp. 897–904, Jul. 2017.

[49] J. Giles, K.K. Ang, L.S. Mihaylova, and M. Arvaneh, "A subject-to-subject transfer learning framework based on Jensen-Shannon divergence for improving brain-computer interface," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Brighton, May 2019, pp. 3087–3091.

[50] L.M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Comput. Math. Math. Phys.*, vol. 7, no. 3, pp. 200–217, 1967.

[51] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning,"*IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 7, pp. 929–942, Jul. 2010.

[52] H. Sun, S. Liu, S. Zhou, and H. Zou, "Unsupervised cross-view semantic transfer for remote sensing image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 13–17, Jan. 2016.

[53] H. Sun, S. Liu, and S. Zhou, "Discriminative subspace alignment for unsupervised visual domain adaptation," *Neural Process. Lett.*, vol. 44, no. 3, pp. 779–793, Dec. 2016.

[54] Q. Shi, Y. Zhang, X. Liu, and K. Zhao, "Regularised transfer learning for hyperspectral image classification," *IET Comput. Vis.*, vol. 13, no. 2, pp. 188–193, Feb. 2019.

[55] A. Gretton, O. Bousquet, A.J. Smola, and B. Schlkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Proc. 18th International Conference on Algorithmic Learning Theory*, Singapore, Oct. 2005, pp. 63–77.

[56] H. Wang and Q. Yang, "Transfer learning by structural analogy," in *Proc. 25th AAAI Conference on Artificial Intelligence*, San Francisco, Aug. 2011, pp. 513–518.

[57] M. Xiao and Y. Guo, "Feature space independent semi-supervised domain adaptation via kernel matching,"*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 54–66, Jan. 2015.

[58] K. Yan, L. Kou, and D. Zhang, "Learning domain-invariant subspace using domain features and independence maximization," *IEEE T. Cybern.*, vol. 48, no. 1, pp. 288–299, Jan. 2018.

[59] J. Shen, Y. Qu, W. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. 32nd AAAI Conference on Artificial Intelligence*, New Orleans, Feb. 2018, pp. 4058–4065.

[60] C.-Y. Lee, T. Batra, M.H. Baig, and D. Ulbricht, "Sliced Wasserstein discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, Jun. 2019, pp. 10285–10295.

[61] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschlger, and S. Saminger-Platz, "Central moment discrepancy (CMD) for domain-invariant representation learning," in *Proc. 5th International Conference on Learning Representations*, Toulon, Apr. 2017, pp. 1–13.

[62] A. Gretton, D. Sejdinovic, H. Strathmann, S. Balakrishnan, M. Pontil, K. Fukumizu, and B.K. Sriperumbudur, "Optimal kernel choice for large-scale two-sample tests," in *Proc. 26th Annual Conference on Neural Information Processing Systems*, Lake Tahoe, Dec. 2012, pp. 1205–1213.

[63] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Jul. 2017, pp. 2272–2281.

[64] H. Daumé III, "Frustratingly easy domain adaptation," in *Proc. 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Jun. 2007, pp. 256–263.

[65] H. Daumé III, A. Kumar, and A. Saha, "Co-regularization based semi-supervised domain adaptation," in *Proc. 24th Annual Conference on Neural Information Processing Systems*, Vancouver, Dec. 2010, pp. 478–486.

[66] L. Duan, D. Xu, and I.W. Tsang, "Learning with augmented features for heterogeneous domain adaptation," in *Proc. 29th International Conference on Machine Learning*, Edinburgh, Jun. 2012, pp. 1–8.

[67] W. Li, L. Duan, D. Xu, and I.W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134–1148, Jun. 2014.

[68] K.I. Diamantaras and S.Y. Kung, *Principal Component Neural Networks*. New York: Wiley, 1996.

[69] B. Schlkopf, A. Smola, and K. Mller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, Jul. 1998.

[70] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced distribution adaptation for transfer learning," in *Proc. 17th IEEE International Conference on Data Mining*, New Orleans, Nov. 2017, pp. 1129–1134.

[71] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annual Conference on Computational Learning Theory*, Madison, Jul. 1998, pp. 92–100.

[72] M. Chen, K.Q. Weinberger, and J.C. Blitzer, "Co-training for domain adaptation," in *Proc. 25th Annual Conference on Neural Information Processing Systems*, Granada, Dec. 2011, pp. 2456–2464.

[73] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, Nov. 2005.

[74] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *Proc. 34th International Conference on Machine Learning*, Sydney, Aug. 2017, pp. 2988–2997.

[75] S.J. Pan, J.T. Kwok, and Q. Yang, "Transfer learning via dimensionality reduction," in *Proc. 23rd AAAI Conference on Artificial Intelligence*, Chicago, Jul. 2008, pp. 677–682.

[76] K.Q. Weinberger, F. Sha, and L.K. Saul, "Learning a kernel matrix for nonlinear dimensionality reduction," in *Proc. 21st International Conference on Machine Learning*, Banff, Jul. 2004, pp. 106–113.

[77] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, no. 1, pp. 49–95, Mar. 1996.

[78] S.J. Pan, I.W. Tsang, J.T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," in *Proc. 21st International Joint Conference on Artificial Intelligence*, Pasadena, Jul. 2009, pp. 1187–1192.

[79] C. Hou, Y.H. Tsai, Y. Yeh, and Y.F. Wang, "Unsupervised domain adaptation with label and structural consistency," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5552–5562, Dec. 2016.

[80] J. Tahmoresnezhad and S. Hashemi, "Visual domain adaptation via transfer feature learning," *Knowl. Inf. Syst.*, vol. 50, no. 2, pp. 585–605, Feb. 2017.

[81] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Jul. 2017, pp. 5150–5158.

[82] B. Schlkopf, R. Herbrich, and A.J. Smola, "A generalized representer theorem," in *Proc. International Conference on Computational Learning Theory*, Amsterdam, Jul. 2001, pp. 416–426.

[83] L. Duan, I.W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[84] A. Rakotomamonjy, F.R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491-2521, Nov. 2008.

[85] I.S. Dhillon, S. Mallela, and D.S. Modha, "Information-theoretic co-clustering," in *Proc. 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, Aug. 2003, pp. 89–98.

[86] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Am. Soc. Inf. Sci.*, vol. 41, pp. 391–407, Sep. 1990.

[87] T. Hofmann, "Probabilistic latent semantic analysis," in *Proc. 15th Conference on Uncertainty in Artificial Intelligence*, Stockholm, Jul. 1999, pp. 289–296.

[88] J. Yoo and S. Choi, "Probabilistic matrix tri-factorization," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Apr. 2009, pp. 1553–1556.

[89] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. - Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[90] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-bridged PLSA for cross-domain text classification," in *Proc. 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, Jul. 2008, pp. 627–634.

[91] F. Zhuang, P. Luo, Z. Shen, Q. He, Y. Xiong, Z. Shi, and H. Xiong, "Collaborative Dual-PLSA: Mining distinction and commonality across multiple domains for text classification," in *Proc. 19th ACM International Conference on Information and Knowledge Management*, Toronto, Oct. 2010, pp. 359–368.

[92] F. Zhuang, P. Luo, Z. Shen, Q. He, Y. Xiong, Z. Shi, and H. Xiong, "Mining distinction and commonality across multiple domains using generative model for text classification," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 11, pp. 2025–2039, Nov. 2012.

[93] F. Zhuang, P. Luo, P. Yin, Q. He, and Z. Shi, "Concept learning for cross-domain text classification: A general probabilistic framework," in *Proc. 23rd International Joint Conference on Artificial Intelligence*, Beijing, Aug. 2013, pp. 1960–1966.

[94] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. Conference on Empirical Methods in Natural Language Processing*, Sydney, Jul. 2006, pp. 120–128.

[95] R.K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *J. Mach. Learn. Res.*, vol. 6, pp. 1817–1853, Dec. 2005.

[96] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. 28th International Conference on Machine Learning*, Bellevue, Jun. 2011, pp. 513–520.

[97] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th International Conference on Machine Learning*, Helsinki, Jul. 2008, pp. 1096–1103.

[98] M. Chen, Z. Xu, K. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," in *Proc. 29th International Conference on Machine Learning*, Edinburgh, Jun. 2012, pp. 767–774.

[99] M. Chen, K.Q. Weinberger, Z. Xu, and F. Sha, "Marginalizing stacked linear denoising autoencoders," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 3849–3875, Jan. 2015.

[100] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *Proc. IEEE International Conference on Computer Vision*, Sydney, Dec. 2013, pp. 2960–2967.

[101] B. Sun and K. Saenko, "Subspace distribution alignment for unsupervised domain adaptation," in *Proc. British Machine Vision Conference*, Swansea, Sep. 2015, pp. 24.1–24.10.

[102] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Providence, Jun. 2012, pp. 2066–2073.

[103] R. Gopalan, Ruonan Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. IEEE International Conference on Computer Vision*, Barcelona, Jun. 2011, pp. 999-1006.

[104] M.I. Zelikin, *Control Theory and Optimization I* in Encyclopaedia of Mathematical Sciences, vol. 86, Berlin: Springer, 2000.

[105] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. 30th AAAI Conference on Artificial Intelligence*, Phoenix, Feb. 2016, pp. 2058–2065.

[106] S.J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-domain sentiment classification via spectral feature alignment," in *Proc. 19th International Conference on World Wide Web*, Raleigh, Apr. 2010, pp. 751–760.

[107] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Jun. 2007, pp. 440–447.

[108] F.R.K. Chung, *Spectral Graph Theory*. Providence: American Mathematical Society, 1997.

[109] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 15th Annual Conference on Neural Information Processing Systems*, Vancouver, Dec. 2001, pp. 849-856.

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

[111] S.D. Kamvar, D. Klein, and C.D. Manning, "Spectral learning," in *Proc. 18th International Joint Conference on Artificial Intelligence*, Acapulco, Aug. 2003, pp. 561–566.

[112] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol .22, no. 8, pp. 888–905, Aug. 2000.

[113] L. Duan, I.W. Tsang, D. Xu, and T.-S. Chua, "Domain adaptation from multiple sources via auxiliary classifiers," in *Proc. 26th International Conference on Machine Learning*, Montreal, Jun. 2009, pp. 289–296.

[114] L. Duan, D. Xu, and I.W. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 504–518, Mar. 2012.

[115] P. Luo, F. Zhuang, H. Xiong, Y. Xiong, and Q. He, "Transfer learning from multiple source domains via consensus regularization," in *Proc. 17th ACM Conference on Information and Knowledge Management*, Napa Valley, Oct. 2008, pp. 103–112.

[116] F. Zhuang, P. Luo, H. Xiong, Y. Xiong, Q. He, and Z. Shi, "Cross-domain learning from multiple sources: A consensus regularization perspective," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 12, pp. 1664–1678, Dec. 2010.

[117] T. Evgeniou, C.A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Mach. Learn. Res.*, vol. 6, pp. 615-637, Apr. 2005.

[118] T. Kato, H. Kashima, M. Sugiyama, and K. Asai, "Multi-task learning via conic programming," in *Proc. 21st Annual Conference on Neural Information Processing Systems*, Vancouver, Dec. 2007, pp. 737–744.

[119] A.J. Smola and B. Schlkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, Aug. 2004.

[120] J. Weston, R. Collobert, F. Sinz, L. Bottou, and V. Vapnik, "Inference with the universum," in *Proc. 23rd International Conference on Machine Learning*, Pittsburgh, Jun. 2006, pp. 1009–1016.

[121] X. Yu and Y. Aloimonos, "Attribute-based transfer learning for object categorization with zero/one training example," in *Proc.*

European Conference on Computer Vision, Heraklion, Sep. 2010, pp. 127–140.

[122] F. Zhuang, P. Luo, H. Xiong, Q. He, Y. Xiong, and Z. Shi, "Exploiting associations between word clusters and document classes for cross-domain text categorization," *Stat. Anal. Data Min.*, vol. 4, no. 1, pp. 100–114, Feb. 2011.

[123] F. Zhuang, P. Luo, C. Du, Q. He, Z. Shi, and H. Xiong, "Triplex transfer learning: Exploiting both shared and distinct concepts for text classification," *IEEE T. Cybern.*, vol. 44, no. 7, pp. 1191–1203, Jul. 2014.

[124] M. Long, J. Wang, G. Ding, W. Cheng, X. Zhang, and W. Wang, "Dual transfer learning," in *Proc. 12th SIAM International Conference on Data Mining*, Anaheim, Apr. 2012, pp. 540–551.

[125] H. Wang, F. Nie, H. Huang, and C. Ding, "Dyadic transfer learning for cross-domain image classification," in *Proc. International Conference on Computer Vision*, Barcelona, Nov. 2011, pp. 551–556.

[126] D. Wang, C. Lu, J. Wu, H. Liu, W. Zhang, F. Zhuang, and H. Zhang, "Softly associative transfer learning for cross-domain classification," *IEEE T. Cybern.*, to be published. doi: 10.1109/TCYB.2019.2891577.

[127] Q. Do, W. Liu, J. Fan, and D. Tao, "Unveiling hidden implicit similarities for cross-domain recommendation," *IEEE Trans. Knowl. Data Eng.*, to be published. doi: 10.1109/TKDE.2019.2923904.

[128] T. Tommasi and B. Caputo, "The more you know, the less you learn: from knowledge transfer to one-shot learning of object categories" in *Proc. British Machine Vision Conference*, London, Sep. 2009, pp. 80.1–80.11.

[129] G.C. Cawley, "Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs," in *Proc. IEEE International Joint Conference on Neural Network*, Vancouver, Jul. 2006, pp. 1661–1668.

[130] T. Tommasi, F. Orabona, and B. Caputo, "Safety in numbers: Learning categories from few examples with multi model knowledge transfer," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, Jun. 2010, pp. 3081–3088.

[131] C.-K. Lin, Y.-Y. Lee, C.-H. Yu, and H.-H. Chen, "Exploring ensemble of models in taxonomy-based cross-domain sentiment classification," in *Proc. 23rd ACM International on Conference on Information and Knowledge Management*, Shanghai, Nov. 2014, pp. 1279–1288.

[132] J. Gao, W. Fan, J. Jiang, and J. Han, "Knowledge transfer via multiple model local structure mapping," in *Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Aug. 2008, pp. 283–291.

[133] F. Zhuang, P. Luo, S.J. Pan, H. Xiong, and Q. He. "Ensemble of anchor adapters for transfer learning," in *Proc. 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis, Oct. 2016, pp. 2335–2340.

[134] F. Zhuang, X. Cheng, P. Luo, S.J. Pan, and Q. He, "Supervised representation learning with double encoding-layer autoencoder for transfer learning," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 2, pp. 1–17, Jan. 2018.

[135] M. Ghifary, W.B. Kleijn, and M. Zhang, "Domain adaptive neural networks for object recognition," in *Proc. 13th Pacific Rim International Conference on Artificial Intelligence*, Gold Coast, Dec. 2014, pp. 898–904.

[136] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, arXiv:1412.3474v1.

[137] M. Long, Y. Cao, J. Wang, and M.I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd International Conference on Machine Learning*, Lille, Jul. 2015, pp. 97–105.

[138] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 26th Annual Conference on Neural Information Processing Systems*, Lake Tahoe, Dec. 2012, pp. 1097–1106.

[139] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 28th Annual Conference on Neural Information Processing Systems*, Montreal, Dec. 2014, pp. 2672–2680.

[140] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. 28th Annual Conference on Neural Information Processing Systems*, Montreal, Dec. 2014, pp. 3320–3328.

[141] M. Long, Y. Cao, Z. Cao, J. Wang, and M.I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published. doi: 10.1109/TPAMI.2018.2868685.

[142] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. 18th Annual Conference on Neural Information Processing Systems*, Vancouver, Dec. 2004, pp. 529–536.

[143] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Boston, Jun. 2015, pp. 1–9.

[144] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Jun. 2016, pp. 770–778.

[145] K.P. Chwialkowski, A. Ramdas, D. Sejdinovic, and A. Gretton, "Fast two-sample testing with analytic representations of probability measures," in *Proc. 29th Annual Conference on Neural Information Processing Systems*, Montreal, Dec. 2015, pp. 1981–1989.

[146] M. Long, H. Zhu, J. Wang, and M.I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. 30th Annual Conference on Neural Information Processing Systems*, Barcelona, Dec. 2016, pp. 136–144.

[147] M. Long, H. Zhu, J. Wang, and M.I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. 34th International Conference on Machine Learning*, Sydney, Aug. 2017, pp. 2208–2217.

[148] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. European Conference on Computer Vision Workshops*, Amsterdam, Oct. 2016, pp. 443–450.

[149] C. Chen, Z. Chen, B. Jiang, and X. Jin, "Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation," in *Proc. 33rd AAAI Conference on Artificial Intelligence*, Honolulu, Jan. 2019, pp. 3296–3303.

[150] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei, "Transferrable prototypical networks for unsupervised domain adaptation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, Jun. 2019, pp. 2239–2247.

[151] G. Kang, L. Jiang, Y. Yang, and A.G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, Jun. 2019, pp. 4893–4902.

[152] Y. Zhu, F. Zhuang, J. Wang, J. Chen, Z. Shi, W. Wu, and Q. He, "Multi-representation adaptation network for cross-domain image classification," *Neural Netw.*, vol. 119. pp. 214–221, Nov. 2019.

[153] Y. Zhu, F. Zhuang, and D. Wang, "Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources," in *Proc. 33rd AAAI Conference on Artificial Intelligence*, Honolulu, Jan. 2019, pp. 5989–5996.

[154] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd International Conference on Machine Learning*, Lille, Jul. 2015, pp. 1180–1189.

[155] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F.Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, pp. 1–35, Apr. 2016.

[156] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Jul. 2017, pp. 2962–2971.

[157] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A.A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proc. 35th International Conference on Machine Learning*, Stockholm, Jul. 2018, pp. 1994–2003.

[158] M. Long, Z. Cao, J. Wang, and M.I. Jordan, "Conditional adversarial domain adaptation," in *Proc. 32nd Annual Conference on Neural Information Processing Systems*, Montreal, Dec. 2018, pp. 1640–1650.

[159] Y. Zhang, H. Tang, K. Jia, and M. Tan, "Domain-symmetric networks for adversarial domain adaptation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, Jun. 2019, pp. 5031–5040.

[160] H. Zhao, S. Zhang, G. Wu, J.M.F. Moura, J.P. Costeira, and G.J. Gordon, "Adversarial multiple source domain adaptation," in *Proc. 32nd Annual Conference on Neural Information Processing Systems*, Montreal, Dec. 2018, pp. 8559–8570.

[161] C. Yu, J. Wang, Y. Chen, and M. Huang, "Transfer learning with dynamic adversarial adaptation network," in *Proc. 19th IEEE International Conference on Data Mining*, Beijing, Nov. 2019, pp. 1–9.

[162] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Jun. 2018, pp. 8156–8163.

[163] Z. Cao, M. Long, J. Wang, and M.I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Jun. 2018, pp. 2724–2732.

[164] B. Wang, M. Qiu, X. Wang, Y. Li, Y. Gong, X. Zeng, J. Huang, B. Zheng, D. Cai, and J. Zhou, "A minimax game for instance based selective transfer learning," in *Proc. 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Anchorage, Aug. 2019, pp. 34–43.

[165] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in *Proc. 36th International Conference on Machine Learning*, Long Beach, Jun. 2019, pp. 1081–1090.

[166] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Miami, Jun. 2009, pp. 248–255.

[167] M. Maqsood, F. Nazir, U. Khan, F. Aadil, H. Jamal, I. Mehmood, and O. Song, "Transfer learning assisted classification and detection of Alzheimer's disease stages using 3D MRI scans," *Sensors*, vol. 19, no. 11, pp. 1–19, Jun. 2019.

[168] D.S. Marcus, A.F. Fotenos, J.G. Csernansky, J.C. Morris, and R.L. Buckner, "Open access series of imaging studies: Longitudinal MRI data in nondemented and demented older adults,"*J. Cogn. Neurosci.*, vol. 22, no. 12, pp. 2677–2684, Dec. 2010.

[169] H.-C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R.M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer Learning," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1285–1298, May 2016.

[170] M. Byra, M. Wu, X. Zhang, H. Jang, Y.-J. Ma, E.Y. Chang, S. Shah, and Jiang Du, "Knee menisci segmentation and relaxometry of 3D ultrashort echo time cones MR imaging using attention UNet with transfer learning," *Magn. Reson. Med.*, Sep. 2019, doi: 10.1002/mrm.27969.

[171] X. Tang, B. Du, J. Huang, Z. Wang, and L. Zhang, "On combining active and transfer learning for medical data classification," *IET Comput. Vis.*, vol. 13, no. 2, pp. 194–205, Feb. 2019.

[172] M. Zeng, M. Li, Z. Fei, Y. Yu, Y. Pan, and J. Wang, "Automatic ICD-9 coding via deep transfer learning," *Neurocomputing*, vol. 324, pp. 43–50, Jan. 2019.

[173] G. Schweikert, G. Ratsch, C. Widmer, and B. Scholkopf, "An empirical analysis of domain adaptation algorithms for genomic sequence analysis," in *Proc. 22nd Annual Conference on Neural Information Processing Systems*, Vancouver, Dec. 2008, pp. 1433–1440.

[174] R. Petegrosso, S. Park, T.H. Hwang, and R. Kuang, "Transfer learning across ontologies for phenome-genome association prediction," *Bioinformatics*, vol. 33, no. 4, pp. 529–536, Feb. 2017.

[175] T. Hwang and R. Kuang, "A heterogeneous label propagation algorithm for disease gene discovery," in *Proc. 10th SIAM International Conference on Data Mining*, Columbus, Apr. 2010, pp. 583–594.

[176] Q. Xu, E.W. Xiang, and Q. Yang, "Protein-protein interaction prediction via collective matrix factorization," in *Proc. IEEE International Conference on Bioinformatics and Biomedicine*, Hong Kong, Dec. 2010, pp. 62–67.

[177] A.P. Singh and G.J. Gordon, "Relational learning via collective matrix factorization," in *Proc. 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, Aug. 2008, pp. 650–658.

[178] S. Di, H. Zhang, C. Li, X. Mei, D. Prokhorov, and H. Ling, "Cross-domain traffic scene understanding: A dense correspondence-based transfer learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 745–757, Mar. 2018.

[179] H. Abdi, "Partial least squares regression and projection on latent structure regression (PLS Regression)," *Wiley Interdiscip. Rev.-Comput. Statist.*, vol. 2, no. 1, pp. 97106, Jan. 2010.

[180] S.D. Pietra, V.D. Pietra, and J. Lafferty, "Inducing features of random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 380–393, Apr. 1997.

[181] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, Dec. 2011.

[182] C. Lu, F. Hu, D. Cao, J. Gong, Y. Xing, and Z. Li, "Transfer learning for driver model adaptation in lane-changing scenarios using manifold alignment," *IEEE Trans. Intell. Transp. Syst.*, to be published. doi: 10.1109/TITS.2019.2925510.

[183] D.J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. Knowledge Discovery in Databases Workshop*, Seattle, Jul. 1994, pp. 359–370.

[184] N. Makondo, M. Hiratsuka, B. Rosman, and O. Hasegawa, "A non-linear manifold alignment approach to robot learning from demonstrations," *J. Robot. Mechatron.*, vol. 30, no. 2, pp. 265–281, Apr. 2018.

[185] P. Angkititrakul, C. Miyajima, and K. Takeda, "Modeling and adaptation of stochastic driver-behavior model with application to car following," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Jun. 2011, pp. 814–819.

[186] Y. Liu, P. Lasang, S. Pranata, S. Shen, and W. Zhang, "Driver pose estimation using recurrent lightweight network and virtual data augmented transfer learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3818–3831, Oct. 2019.

[187] J. Wang, H. Zheng, Y. Huang, and X. Ding, "Vehicle type recognition in surveillance images from labeled web-nature data using deep transfer learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2913–2922, Sep. 2018.

[188] K. Gopalakrishnan, S.K. Khaitan, A. Choudhary, and A. Agrawal, "Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection," *Constr. Build. Mater.*, vol. 157, pp. 322–330, Dec. 2017.

[189] S. Bansod and A. Nandedkar, "Transfer learning for video anomaly detection," *J. Intell. Fuzzy Syst.*, vol. 36, no. 3, pp. 1967–1975, Mar. 2019.

[190] G. Rosario, T. Sonderman, and X. Zhu, "Deep transfer learning for traffic sign recognition," in *Proc. IEEE International Conference on Information Reuse and Integration*, Salt Lake City, Jul. 2018, pp. 178–185.

[191] W. Pan, E.W. Xiang, and Q. Yang, "Transfer learning in collaborative filtering with uncertain ratings," in *Proc. 26th AAAI Conference on Artificial Intelligence*, Toronto, Jul. 2012, pp. 662–668.

[192] G. Hu, Y. Zhang, and Q. Yang, "Transfer meets hybrid: A synthetic approach for cross-domain collaborative filtering with text," in *Proc. 28th International Conference on World Wide Web*, San Francisco, May 2019, pp. 2822–2829.

[193] W. Pan, E.W. Xiang, N.N. Liu, and Q. Yang, "Transfer learning in collaborative filtering for sparsity reduction," in *Proc. 24th AAAI Conference on Artificial Intelligence*, Atlanta, Jul. 2010, pp. 230–235.

[194] W. Pan and Q. Yang, "Transfer learning in heterogeneous collaborative filtering domains," *Artif. Intell.*, vol. 197, pp. 39–55, Apr. 2013.

[195] F. Zhuang, Y. Zhou, F. Zhang, X. Ao, X. Xie, and Q. He, "Sequential transfer learning: Cross-domain novelty seeking trait mining for recommendation," in *Proc. 26th International Conference on World Wide Web Companion*, Perth, Apr. 2017, pp. 881–882.

[196] J. Zheng, F. Zhuang, and C. Shi, "Local ensemble across multiple sources for collaborative filtering," in *Proc. 26th ACM International on Conference on Information and Knowledge Management*, Singapore, Nov. 2017, pp. 2431–2434.

[197] F. Zhuang, J. Zheng, J. Chen, X. Zhang, C. Shi, and Q. He, "Transfer collaborative filtering from multiple sources via consensus regularization," *Neural Netw.*, vol. 108, pp. 287–295, Dec. 2018.

[198] J. He, R. Liu, F. Zhuang, F. Lin, C. Niu, and Q. He, "A general cross-domain recommendation framework via Bayesian neural network," in *Proc. 18th IEEE International Conference on Data Mining*, Singapore, Nov. 2018, pp. 1001–1006.

[199] F. Zhu, Y. Wang, C. Chen, G. Liu, M.A. Orgun, and J. Wu, "A deep framework for cross-domain and cross-system recommendations," in *Proc. 27th International Joint Conference on Artificial Intelligence*, Stockholm, Jul. 2018, pp. 3711–3717.

[200] F. Yuan, L. Yao, and B. Benatallah, "DARec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns," in *Proc. 29th International Joint Conference on Artificial Intelligence*, Macao, Aug. 2019, pp. 4227–4233.

[201] E. Bastug, M. Bennis, and M. Debbah, "A transfer learning approach for cache-enabled wireless networks," in *Proc. 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, Mumbai, May 2015, pp. 161–166.

[202] R. Li, Z. Zhao, X. Chen, J. Palicot, and H. Zhang, "TACT: A transfer actor-critic learning framework for energy saving in cellular radio access networks," *IEEE Trans. Wirel. Commun.*, vol. 13, no. 4, pp. 2000–2011, Apr. 2014.

[203] Q. Zhao and D. Grace, "Transfer learning for QoS aware topology management in energy efficient 5G cognitive radio networks," in

Proc. 1st International Conference on 5G for Ubiquitous Connectivity, Akaslompolo, Nov. 2014, pp. 152–157.

[204] B. Guo, J. Li, V.W. Zheng, Z. Wang, and Z. Yu, "Citytransfer: Transferring inter- and intra-city knowledge for chain store site recommendation based on multi-source urban data," in *Proc. ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, Jan. 2018, pp. 1–23.

[205] Y. Wei, Y. Zheng, and Q. Yang, "Transfer knowledge between cities," in *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, Aug. 2016, pp. 1905–1914.

[206] U. Cote-Allard, C.L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin, "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760–771, Apr. 2019.

[207] C. Ren, D. Dai, K. Huang, and Z. Lai, "Transfer learning of structured representation for face recognition," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5440–5454, Dec. 2014.

[208] J. Wang, Y. Chen, L. Hu, X. Peng, and P.S. Yu, "Stratified transfer learning for cross-domain activity recognition," in *Proc. IEEE International Conference on Pervasive Computing and Communications*, Athens, Mar. 2018, pp. 1–10.

[209] J. Deng, Z. Zhang, E. Marchi, and B. Schuller, "Sparse autoencoder-based feature transfer learning for speech emotion recognition," in *Proc. Humaine Association Conference on Affective Computing and Intelligent Interaction*, Geneva, Sep. 2013, pp. 511–516.

[210] D. Xi, F. Zhuang, G. Zhou, X. Cheng, F. Lin, and Q. He, "Domain adaptation with category attention network for deep sentiment analysis," in *Proc. The Web Conference*, Taipei, Apr. 2020, pp. 3133–3139.

[211] Y. Zhu, D. Xi, B. Song, F. Zhuang, S. Chen, X. Gu, and Q. He, "Modeling users' behavior sequences with hierarchical explainable network for cross-domain fraud detection," in *Proc. The Web Conference*, Taipei, Apr. 2020, pp. 928–938.

[212] J. Tang, T. Lou, J. Kleinberg, and S. Wu, "Transfer learning to infer social ties across heterogeneous networks," *ACM Trans. Inf. Syst.*, vol. 34, no. 2, pp. 1–43, Apr. 2016.

[213] L. Zhang, L. Zhang, D. Tao, and X. Huang, "Sparse transfer manifold embedding for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sensing*, vol. 52, no. 2, pp. 1030–1043, Feb. 2014.

[214] F. Zhuang, K. Duan, T. Guo, Y. Zhu, D. Xi, Z. Qi, and Q. He, "Transfer learning toolkit: Primers and benchmarks," 2019, arXiv:1911.08967v1.

[215] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. 11th European Conference on Computer Vision*, Heraklion, Sep. 2010, pp. 213–226.

[216] Z.C. Lipton, "The mythos of model interpretability," *ACM Que.*, vol. 16, no. 3, May 2018, pp. 1–27.