Transfer Learning using Computational Intelligence: A Survey

Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, Guangquan Zhang
Decision Systems & e-Service Intelligence (DeSI) Lab, Centre for Quantum Computation & Intelligent Systems (QCIS), Faculty of Engineering and Information Technology,
University of Technology Sydney, POBOX123, Broadway, NSW2007, Australia
jie.lu@uts.edu.au, vahid.behbood@uts.edu.au, peng.hao@student.uts.edu.au, hua.zuo@student.uts.edu.au, shan.xue@student.uts.edu.au, guangquan.zhang@uts.edu.au

Abstract

Transfer learning aims to provide a framework to utilize previously-acquired knowledge to solve new but similar problems much more quickly and effectively. In contrast to classical machine learning methods, transfer learning methods exploit the knowledge accumulated from data in auxiliary domains to facilitate predictive modeling consisting of different data patterns in the current domain. To improve the performance of existing transfer learning methods and handle the knowledge transfer process in real-world systems, computational intelligence has recently been applied in transfer learning. This paper systematically examines computational intelligence-based transfer learning techniques and clusters related technique developments into four main categories: a) neural network-based transfer learning; b) Bayes-based transfer learning; c) fuzzy transfer learning, and d) applications of computational intelligence-based transfer learning. By providing state-of-the-art knowledge, this survey will directly support researchers and practice-based professionals to understand the developments in computational intelligence-based transfer learning research and applications.

Keywords: Transfer learning, computational intelligence, neural network, Bayes, fuzzy sets and systems, genetic algorithm.

1. Introduction

Although machine learning technologies have attracted a remarkable level of attention from researchers in different computational fields, most of these technologies work under the common assumption that the training data (source domain) and the test data (target domain) have identical feature spaces with underlying distribution. As a result, once the feature space or the feature distribution of the test data changes, the prediction models cannot be used and must be rebuilt and retrained from scratch using newly-collected training data, which is very expensive and sometimes not practically possible. Similarly, since learning-based models need adequate labeled data for training, it is nearly impossible to establish a learning-based model for a target domain which has very few labeled data available for supervised learning. If we can transfer and exploit the knowledge from an existing similar but not identical source domain with plenty of labeled data, however, we can pave the way for construction of the learning-based model for the target domain. In real world scenarios, there are many situations in which very few labeled data are available, and collecting new labeled training data and forming a particular model are practically impossible.

Transfer learning has emerged in the computer science literature as a means of transferring knowledge from a source domain to a target domain. Unlike traditional machine learning and semi-supervised algorithms [1-4], transfer learning considers that the domains of the training data and the test data may be different [5]. Traditional machine learning algorithms make predictions on the future data using mathematical models that are trained on previously collected labeled or unlabeled training data which is the same as future data [6-8]. Transfer learning, in contrast, allows the domains, tasks, and distributions used in training and testing to be different. In the real world, we observe many examples of transfer learning. We may find that learning to recognize apples might help us to recognize pears, or learning to play the electronic organ may facilitate learning the piano. The study of transfer learning has been inspired by the fact that human beings can utilize previously-acquired knowledge to solve new but similar problems much more quickly and effectively. The fundamental motivation for transfer learning in the field of machine learning focuses on the need for lifelong machine learning methods that retain and reuse previously learned knowledge. Research on transfer learning has been undertaken since 1995 under a variety of names: learning to learn; life-long learning; knowledge transfer; meta learning; inductive transfer; knowledge consolidation; context sensitive learning and multi-task learning [9]. In 2005, the Broad Agency Announcement of the Defense Advanced Research Projects Agency's Information Processing Technology Office gave a new mission to transfer learning: the ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks. In this definition, transfer learning aims to extract the knowledge from one or more source tasks and then apply the knowledge to a target task. Traditional machine learning techniques only try to learn each task from scratch, while transfer learning techniques try to transfer the knowledge from other tasks and/or domains to a target task when the latter has few high-quality training data.

Several survey papers on transfer learning have been published in the last few years. For example, the paper by [9] presented an extensive overview of transfer learning and different categories. However, these papers focus on transfer learning techniques and approaches only; none of them discusses how the computational intelligence approach can be used in transfer learning. Since the computational intelligence approach has been applied in transfer learning more recently and has already demonstrated its advantage, this survey is timely.

There are three main types of articles being reviewed in this survey: Type 1 — articles on transfer learning techniques (including related methods and approaches) and Type 2 — articles on transfer learning using computational intelligence techniques. Type 3 — articles on related computational intelligence techniques. The search and selection of these articles were performed according to the following five steps:

Step 1. Publication database identification and determination: The eminent publication databases such as Science Direct, ACM Digital Library, IEEE Xplore and SpringerLink, were searched to provide a comprehensive bibliography of research papers on transfer learning and transfer learning using computational intelligence.

Step 2. Type 1 article selection: These papers were selected according to the two criteria: 1) novelty; 2) impact-published in high quality (high impact factor) journals, or in conference proceedings or book chapters but with high citations[1]. These types of article are mainly used in Section 2.

Step 3. Preliminary screening of Type 2 articles: The search was first performed based on related keywords of computational intelligence in transfer learning.

Step 4. Result filtering for Type 2 articles: The keywords of the preliminary references were extracted and clustered manually. Based on the keywords related to application domain, these papers were divided, using "topic clustering", into four groups: a) Neural Network in transfer learning; b) Bayes in transfer learning; c) fuzzy and genetic algorithm in transfer learning and d) application of transfer learning. This article selection process was based on the following criteria: 1) novelty — published within the last few years; 2) impact — see Step 2; 3) coverage — reported a new or particular application domain; 4) typicality — only the most typical methodology and applications were retained.

Step 5. Type 3 article selection: These papers were selected according to the requirement of Step 4, aiming to introduce related concepts of computational intelligence techniques.

The main contributions of this paper are: 1) it comprehensively and perceptively summarizes research achievements on transfer learning from the point of view of applications of computational intelligence, and strategically clusters the transfer learning into four computational intelligence application domains; 2) for each computational intelligence technique it carefully analyses typical transfer learning frameworks and effectively identifies the specific requirements of computational intelligence techniques in transfer learning. This will directly support researchers and practitioners to promote the popularization and application of computational intelligence in transfer learning in different domains; 3) it also covers several very new transfer learning techniques with computational intelligence, and reveals their successful applications.

The remainder of this paper is structured as follows. In Section 2, the transfer learning techniques are reviewed and analyzed. Sections 3 to 5 respectively present the 4 main application domains of transfer learning. Section 6 discusses the applications of computational intelligence-based transfer learning methods. Section 7 presents our analysis and main findings.

2. Basic transfer learning techniques

To understand and analyze the application developments of transfer learning by using computational intelligence, this section first reviews the main transfer learning techniques. The notations and definitions that will be used throughout the section are introduced. According to the definitions, we then categorize the various settings of transfer learning methods that exist in the literature of machine learning.

Definition 2.1 (Domain) [9] A domain, which is denoted by $D = \{\chi, P(X)\}$, consists of two components:

(1) Feature space χ; and

(2) Marginal probability distribution $P(X)$, where $X = \{x_1, \dots, x_n\} \in \chi$.

Definition 2.2 (Task) [9] A task, which is denoted by $T = \{Y, f(\cdot)\}$, consists of two components:

(1) A label space $Y = \{y_1, \dots, y_m\}$; and

(2) An objective predictive function$f(\cdot)$which is not observed and is to be learned by pairs $\{x_i, y_i\}$.

The function $f(\cdot)$ can be used to predict the corresponding label, $f(x_i)$, of a new instance x_i. From a probabilistic viewpoint, $f(x_i)$ can be written as $P(y_i|x_i)$. In the bank failure prediction example, which is a binary prediction task, y_i can be the label of failed or survived. More specifically, the source domain can be denoted as $D_s = \{(x_{s_1}, y_{s_1}), \dots, (x_{s_n}, y_{s_n})\}$ where $x_{s_i} \in \chi_s$ is the source instance or bank in the bank failure prediction example and $y_{s_i} \in Y_s$ is the corresponding class label which can be failed or survived for bank failure prediction. Similarly, the target domain can be denoted as $D_t = \{(x_{t_1}, y_{t_1}), \dots, (x_{t_n}, y_{t_n})\}$ where $x_t \in \chi_t$ is the target instance and $y_{t_i} \in Y_t$ is the corresponding class label and in most scenarios $t_n \ll s_n$.

Definition 2.3 (Transfer learning) [9] Given a source domain D_s and learning task T_s, a target domain D_t and learning task T_t, transfer learning aims to improve the learning of the target predictive function $f_t(\cdot)$ in D_t using the knowledge in D_s and T_s where $D_s \neq D_t$ or $T_s \neq T_t$.

In the above definition, the condition $D_s \neq D_t$ implies that either $\chi_s \neq \chi_t$ or $P_s(X) \neq P_t(X)$. Similarly, the condition $T_s \neq T_t$ implies that either $Y_s \neq Y_t$ or $f_s(\cdot) \neq f_t(\cdot)$. In addition, there are some explicit or implicit relationships between the feature spaces of two domains such that we imply that the source domain and target domain are related. It should be mentioned that when the target and source domains are the same ($D_s = D_t$) and their learning tasks are also the same ($T_s = T_t$), the learning problem becomes a traditional machine learning problem.

According to the uniform definition of transfer learning introduced by Definition 2.3, transfer learning techniques can be divided into three main categories [9]: 1) *Inductive transfer learning*, in which the learning task in the target domain is different from the target task in the source domain ($T_s \neq T_t$); 2) *Unsupervised transfer learning* which is similar to inductive transfer learning but focuses on solving unsupervised learning tasks in the target domain such as clustering, dimensionality reduction and density estimation ($T_s \neq T_t$); and 3) *Transductive transfer learning*, in which the learning tasks are the same in both domains, while the source and target domains are different ($T_s = T_t,\ D_s \neq D_t$). In the literature, transductive transfer learning, domain adaptation, covariate shift, sample selection bias, transfer learning, multi-task learning, robust learning, and concept drift are all terms which have been used to handle the related scenarios. More specifically, when the method aims to optimize the performance on multiple tasks or domains simultaneously, it is considered to be multi-task learning. If it optimizes performance on one domain, given training data that is from a different but related domain, it is considered to be transductive transfer learning or domain adaptation. Transfer learning and transductive transfer learning have often been used interchangeably with domain adaptation. Concept drift refers to a scenario in which data arrives sequentially with changing distribution, and the goal is to predict the next batch given the previously-arrived data [10].The goal of robust learning is to build a classifier that is less sensitive to certain types of change, such as feature change or deletion in the test data. In addition, unsupervised domain adaptation can be considered as a form of semi-supervised learning, but it assumes that the labeled training data and the unlabeled test data are drawn from different distributions. The existing techniques and methods, which have thus far been used to handle the domain adaptation problem, can be divided into four main classes [11]:

1) Instance weighting for covariate shift methods which weight samples in the source domain to match the target domain. The covariate shift scenario might arise in cases where the training data has been biased toward one region of the input space or is selected in a non-I.I.D. manner. It is closely related to the idea of sample-selection bias which has long been studied in statistics [12] and in recent years it has been explored for machine-learning. Huang et al. [13] proposed a novel procedure called Kernel Mean Matching (KMM) to estimate weights on each instance in the source domain, based on the goal of making the weighted distribution of the source domain look similar to the distribution of the target domain. Sugiyama et al. [14] and Tsuboi et al. [15] proposed a similar idea called the Kullback-Leibler Importance Estimation Procedure (KLIEP). Here too the goal is to estimate weights to maximize similarity between the target and weight-corrected source distributions.

2) Self-labeling methods which include unlabeled target domain samples in the training process and initialize their labels and then iteratively refine the labels. Self-training has a close relationship with the Expectation Maximization (EM) algorithm, which has hard and soft versions. The hard version adds samples with single certain labels while the soft version assigns label confidences when fitting the model. Tan et al. [16] modified the relative contributions of the source and target domains in EM. They increased the weight on the target data at each iteration, while Dai et al. [17] specified the tradeoff between the source and target data terms by estimating KL divergence between the source and target distributions, placing more weight on the target data as KL divergence increases. Self-training methods have been applied to domain adaptation on Natural Language Processing (NLP) tasks including parsing [18-21]; part-of-speech tagging [22]; conversation summarization [23]; entity recognition [22, 24, 25]; sentiment classification [26]; spam detection [22]; cross-language document classification [27, 28]; and speech act classification [29].

3) Feature representation methods which try to find a new feature representation of the data, either to make the target and source distributions look similar, or to find an abstracted representation for domain-specific features. The feature representation approaches can be categorized into two classes [11]: *(A) Distribution similarity* approaches aim explicitly to make the source and target domain sample distributions similar, either by penalizing or removing features whose statistics vary between domains [24, 30-32] or by learning a feature space projection in which a distribution divergence statistic is minimized [33-35]; *(B) Latent feature* approaches aim to construct new features by analyzing large amounts of unlabeled source and target domain data [25, 36-42].

4) Cluster-based learning methods rely on the assumption that samples connected by high-density paths are likely to have the same label if there is a high density path between them [43]. These methods aim to construct a graph in which the labeled and unlabeled samples are the nodes, with the edge weights among samples based on their similarity. Dai et al. [17] proposed a co-clustering based algorithm to propagate the label information across domains for document classification. Xue et al. [44] proposed a cross-domain text classification algorithm known as TPLSA to integrate labeled

and unlabeled data from different but related domains.

3. Transfer learning using neural network

Neural Network aims to solve complex non-linear problems using a learning-based method inspired by human brain structure and processes. In classical machine learning problems, many studies have demonstrated the superior performance of neural network compared to statistical methods. This fact has encouraged many researchers to use neural network for transfer learning, particularly in complicated problems. To address the problem in transfer learning, a number of neural network-based transfer learning algorithms have been developed in recent years. This section reviews three of the principal *Neural Network* techniques: *Deep Neural Network*, *Multiple Tasks Neural Network*, and *Radial Basis Function Neural Network*, and presents their applications in transfer learning.

3.1. Transfer learning using deep neural network

Deep neural network is considered to be an intelligent feature extraction module that offers great flexibility in extracting high-level features in transfer learning. The prominent characteristic of deep neural network is its multiple hidden layers, which can capture the intricate non-linear representations of data. Hubel and Wiesel [45] proposed multi-stage Hubel-Wiesel architectures that consist of alternating layers of convolutions and max pooling to extract data features. A new model blending the above structure and multiple tasks is proposed for transfer learning [46]. In this model, a target task and related tasks are trained together with shared input and hidden layers, and separately output neurons. The model is then extended to the case in which each task has multiple output neurons [47]. Likewise, based on the multi-stage Hubel-Wiesel architectures, whether shared hidden layers trained by the source task can be reused on a different target task is detected. For the target task model, only the last classification layer needs to be retrained, but any layer of the new model could be fine-tuned if desired. In this case, the parameters of hidden layers in the source task model act as initialization parameters of the new target task model, and this strategy is especially promising for a model in which good initialization is very important [48]. As mentioned above, generally all the layers except the last layer are treated as feature extractors in a deep neural network. In contrast to this network structure, a new feature extractor structure is proposed by Collobert and Weston [49]. Only the first two layers are used to extract features at different levels, such as word level and sentence level in Natural Language Processing. Subsequent layers are classical neural network layers used for prediction. The Stacked Denoising Autoencoder (SDA) is another structure that is presented in deep neural network [50]. The core idea of SDA is that unsupervised learning is used to pre-train each layer, and ultimately all layers are fine-tuned in a supervised learning way. Based on the SDA model, different feature transference strategies are introduced to target tasks with varying degrees of complexity. The number of layers transferred to the new model depends on the high-level or low-level feature representations that are needed. This means if low-level features are needed, only the first layer parameters are transferred to the target task [50]. In addition, an interpolating path is presented to transfer knowledge from the source task to the target task in a deep neural network. The original high dimensional features of the source and target domains are projected to lower dimensional subspaces that lie on the Grassman manifold, which presents a way to interpolate smoothly between the source and target domains; thus, a series of feature sets is generated on the interpolating path and intermediate feature extractors are formed based on deep neural network [51]. Deep neural networks can also combine with other technology to promote the performance of transfer learning. Swietojanski, Ghoshal [52] applied restricted Boltzmann machine to pre-train deep neural network, and the outputs of the network are used as features for a hidden Markov model.

3.2. Transfer learning using multiple task neural network

To improve the learning for the target task, multiple task learning (MTL) is proposed. Information contained in other related tasks is used to promote the performance of target task [53]. In multiple task neural network learning, all tasks are trained in parallel using the shared input and hidden neurons and separate output neurons depending on different tasks [54]. The biggest difference between the MTL here and the MTL in deep neural network is the number of hidden layers. Generally, the number of hidden layers in MTL is much smaller than in deep neural networks. In MTL, source tasks as auxiliary information help the target task to improve performance. However, due to different relatedness between source tasks and the target task, the contributions of source tasks should be distinguished. Therefore, a modified version of multitask learning called ηMTL is introduced. Based on a measure of relatedness between source tasks and the target task, ηMTL applies a separate learning rate for each task output neuron [55]. Silver and Mercer [56] presented a task rehearsal method (TRM) to transfer knowledge of source tasks to the target task at a functional level. Instead of the interrelation between representations of various tasks, the relationship between functions of tasks is the core content in their new model. After demonstrating the good performance of ηMTL and TRM on synthetic tasks, they were practically applied to a series of medical diagnostic tasks [57]. In the MTL model, the output layer consists of a separate neuron corresponding to each task, which may lead to redundant outputs and overlapping information. In addition, for the continuous tasks, contextual cues must be provided to guide the system to associate an example with a particular task. In light of these problems, Silver and Poirier [58] proposed context-sensitive multiple task learning (csMTL) with two major differences. To eliminate the redundant outputs and reduce the free parameters, only one neuron is included in the output layer. Another difference is reflected in the input layer, which can be divided into two parts. Apart from the set of input variables for the tasks, the input layer also contains a set of context inputs that associates each training example with a particular task. To verify the effectiveness of csMTL, a set of experiments was designed to detect csMTL and MTL neural networks in their ability to transfer knowledge [59]. The above model makes the assumption that each task only has one output neuron. Further, csMTL is extended to learn tasks that have multiple output neurons [60].

3.3. *Transfer learning using radial basis function neural networks*

Yamauchi [61] considered covariate shift, one category of transfer learning, and incremental learning. Under the assumption that incremental learning environments are a subset of covariate shift, a novel incremental learning method is presented on the basis of radial basis function neural network. Further, a number of model-selection criteria are set up to optimize the network; for example, the information criterion [62] is applied to determine the number of hidden neurons [63]. In some literatures, neural network acts as a part of the whole algorithm. Liu et al. [64] applied neural network to initialize the weights of labeled data in the source domain. Each instance in the source domain is input into the neural network trained by limited target labeled data to gain the contribution degree based on the error value. In addition, the neural network is used as pre-processing technique to extract features from high dimensional space to low dimensional space [65]. Sometimes, neural network is combined with other intelligent techniques to form an integrated model to improve the performance of transfer learning [66].

4. Transfer learning using Bayes

Bayesian techniques refer to methods that are related to statistical inference and are developed based on Bayesian theorem. A Bayesian classifier is a probabilistic methodology for solving classification problems. Since probability is a useful tool for modeling the uncertainty in the real world and is adequate for quantifying the certainty degree of an uncertain truth, Bayesian classifier is popular in the machine learning community. When it comes to the transfer learning setting, the distribution of the training data and test data is not identical, so a Bayesian classifier trained on training data may not be predictive for the test data. To address this challenging problem, many Bayesian-based transfer learning algorithms have been developed in recent years. This section reviews three of the main Bayesian techniques: naïve Bayes classifier, Bayesian network and the hierarchical Bayesian model, and illustrates their application within the framework of transfer learning.

4.1. Transfer learning using Naïve Bayes

The naïve Bayes classifiers [67] are among the most popular classifiers in real world application. They pose a simple but strong assumption that there is independence between each pair of features (attributes) given the class variables. Though this assumption is not suitable in most real scenarios, naïve Bayes classifiers have nevertheless been proved to work quite well in some complicated applications, especially automatic medical diagnosis [68], spam filtering [69] and text categorization [70], in which they may even outperform more advanced algorithms, such as support vector machine, or random forests. Normally, the probabilistic model for a classifier is

$$p(C|F_1,\cdots,F_n)=\frac{p(C)p(F_1,\cdots,F_n|C)}{p(F_1,\cdots,F_n)} \tag{1}$$

where $p(C|F_1,\cdots,F_n)$ indicates a posteriori probability of class variable C, conditional on feature variables F_1 through F_n. Since $p(F_1,\cdots,F_n)$ has no relation with the class variable and the value of F_i (i = 1,..,n) is observable, the above equation can be expressed as

$$p(C|F_1,\cdots,F_n)\propto p(C)p(F_1,\cdots,F_n|C) \tag{2}$$

Under the independence assumption adopted by naïve Bayes classifier, which means

$$p(C|F_1,\cdots,F_n)\propto p(C)\prod_{i=1}^{n}p(F_i|C) \tag{3}$$

From equation (3) we find that a prediction made by a classifier depends on the prior probability of the class variable and the product of the likelihood of each feature variable given a specific class variable. To estimate each feature's distribution, it is necessary to make parameter estimation, assuming a predefined distribution (i.e., multinomial distribution or multivariate Bernoulli distribution) or generating a non-parametric model for a feature that comes from training data. However, if the test data (new-domain data) follow a different distribution from the training data (old-domain data), we cannot obtain an accurate feature distribution estimation for the new-domain data based on the parameter learned from the old-domain data, which leads to bad prediction performance in the result. Estimating the feature distribution for new-domain unlabeled data limits the application of the naïve Bayes classifier in the transfer learning setting.

To adapt the naïve Bayes classifier from the training data to the test data, [17] proposed a novel naïve Bayes transfer learning (NBTL) classification algorithm for text categorization. NBTL first trains a naïve Bayes classifier on the training data and applies the learned classifier on the test data to obtain a pseudo label for the test data during learning, thereby providing an initial model estimation for the test data under target distribution. The Expectation-Maximization (EM) algorithm is then applied in iteration to find a local optimal model only for fitting the target distribution, meaning that the naïve Bayes classifier trained on the training data is adapted to the test data. To measure the difference between the different distributions, KL divergence is used to estimate a trade-off parameter in the NBTL model, and the experiment results show that the performance of NBTL increases when the distribution between the training data and the test data is significantly different. The main disadvantage of NBTL lies in the fact that the influence of new-domain specific features is ignored. Instead of treating both old-domain and new-domain data equally, an adaptive naïve Bayes is proposed in [16]. It uses a weighted EM algorithm to dynamically increase the importance of new-domain data and decrease the weight of old data, while at the same time emphasizing the usage of both generalizable features drawn from the old-domain data and all the features from the new-domain data for tackling the cross-domain sentiment classification problem. Roy and Kaelbling [71] developed an alternative method of transferring the naïve Bayes classifier. They first partition the dataset into a number of clusters, such that the data for each cluster for all tasks has the same distribution. Then they train one classifier for each partition; all classifiers are then combined using a Dirichlet process.

In addition to text classification, [72] developed a transfer naïve Bayes (TNB) algorithm to predict cross-company software defects. The implementation can be summarized in three steps: it first collects maximum and minimum value vectors of the target feature from test data, then each feature of a training sample is compared with the corresponding part of those two vectors to calculate the number of similar attributes and the weight of that training instance is computed through a gravitational analogy. After obtaining all the weights for the training data, a prediction model can be built with those weighted training data to classify the test dataset.

4.2. Transfer learning using Bayesian network

Assuming total independence between features is not applicable for many real world problems, because the occurrence of an event may arise as the result of a number of relevant factors. In other words, there are correlations between features in a decision region and the Bayesian network is a suitable representation to this fact. A Bayesian network is a graphical model that encodes probabilistic relationships among variables of interest. It consists of two components: (1) a directed acyclic graph (DAG), which contains nodes and arcs. In particular, the nodes can be observed quantities, latent variables, or unknown parameters, while the directed arcs reflect conditional dependencies among variables, and (2) conditional probability tables (CPTs), which record local probability distributions associated with each node. Bayesian networks have four distinct advantages when compared to other data mining methods, namely, the ability to handle incomplete datasets, to discover causal relationships hidden in the data, to incorporate both domain knowledge and data into one model, and to avoid data over-fitting [73].

In a simple case, the graphical model of a Bayesian network can be constructed by the prior knowledge of an expert. However in some complex applications, the definition of "network" is difficult for humans, so it is necessary to learn the network structure and parameters of the local distributions from data [74]. To learn a Bayesian network from data, one needs to consider two important phases: structure learning and parameter learning, respectively. The former relates to the learning of a graphical model from data, while the latter deals with the evaluation of condition probability distribution for each variable given the model. To our knowledge, most works focus on structure learning by leveraging previous data, and less effort is expended on parameter learning.

When the training data in a task is scarce, learning a reliable Bayesian network is difficult; therefore transfer learning can help improve the robustness of learned networks through exploiting data from related tasks or knowledge from domain experts. In [75], the authors extended the Bayesian network learning from a single domain (task) to multiple domains (tasks). In this case, instead of learning a structure in isolation, the relationships between tasks should be taken into account. Similar to the multi-task learning scenario, multiple Bayesian network structures are jointly built from multiple related datasets. To make learning efficient, the parameters of Bayesian networks from related tasks are assumed to be independent. The prior is defined in such a way that it penalizes structures that are different from one another. A score and search approach is then performed on the space of multiple Bayesian networks to find the best structures, in particular, by defining a suitable search space and devising a branch and bound procedure that enables efficient moves in this search space. In contrast to learning optimal models simultaneously for different tasks, [76] proposed learning models from auxiliary tasks to improve related tasks. In this paper, without giving sufficient data for independence test, a PC-TL algorithm is developed with consideration of both the confidence of the independence tests and the similarity of the auxiliary tasks to the target task in a combined function. An example that uses transfer learning to strengthen the quality of learned Bayesian networks through the use of an inductive bias may also be found in [77]. The main limitation of such multi-task network structure learning algorithms lies in the assumption that all pairs of tasks are equally related, which violates the truth that different pairs of tasks can differ in their degree of relatedness. As a result, Oyen and Lane [78] relaxed this assumption by adding a task relatedness metric, which explicitly controls the amount of information sharing between tasks, into a learning objective to incorporate domain knowledge about task-relatedness. Experimental results show that leveraging domain knowledge produces models that are both robust and in accordance with a domain expert's objective. Recently, Oyen and Lane [79] pointed out that it is more appropriate to estimate a posterior distribution over multiple learned Bayesian networks rather than a single posteriori. In their paper, the authors proposed the incorporation of structure bias into order-conditional network discovery algorithms to extend network discovery in individual Bayesian network learning [80, 81] for transfer network learning.

Given a Bayesian network structure, the work of parameter learning is to estimate the conditional probability tables (CPTs) for each node given the combination of its parent's nodes. If we have data from all tasks, then we can directly estimate the CPTs from data. However in some cases we only have models from related tasks and need to estimate the CPT for the target task. In [76], two novel aggregation methods were defined. The first calculates a weighted average of the probabilities from the data of the auxiliary tasks based on the confidence of the probability estimated from the auxiliary tasks and the similarity with the target estimates. This average is then combined with the target probability estimate, weighted by a factor that depends on its similarity to the target probability. The second method works similarly, but the average of probabilities is obtained from those closer to the target rather than from all the data of the auxiliary tasks. In addition, the average is combined to the target estimate with a confidence factor, which is based on the amount of data.

4.3. Transfer learning using hierarchical Bayesian model

Hierarchical Bayesian models are considered to be a particular type of Bayesian network and are used when the data are structured in groups. This hierarchical model can represent and reason about knowledge at multiple levels of abstraction, therefore a hierarchical Bayesian model provides a unified framework to explain both how abstract knowledge is used for induction and how abstract knowledge can be acquired.

In considering the problem of multi-task learning, Wilson et al. [82] used a hierarchical Bayesian infinite mixture model to model the distribution over multiple Markov Decision Processes (MDPs) such that the characteristics of new environments can be quickly inferred based on the learned distribution as an informed prior. This idea is extended to solve the problem of sequential decision-making tasks [83]. Yang et al. [84] combined all the tasks in a single RBF network and defined a novel Bayesian multi-task learning model for non-linear regression. Meanwhile, Raykar et al. [85] presented a novel Bayesian multiple instance learning (MIL) algorithm, which performs feature selection and classifier construction simultaneously. The results show that the proposed method is more accurate and effective when a smaller set of useful features is selected.

In reference to the domain adaptation problem, a novel hierarchical Bayesian domain adaptation model was developed based on the use of a hierarchical Bayes prior [86]. In the proposed model, several parameters are set to each feature in each domain, and top level parameters are proposed on the upper level such that the Gaussian prior over the parameter values in each domain is now centered around these top level parameters instead of around zero, while the zero-mean Gaussian prior is placed over the top level parameters. At the same time, Wood and Teh [87] designed a doubly hierarchical Pitman-Yor process language model, in which the bottom layer utilizes multiple hierarchical Pitman-Yor process language models to represent a number of domains while the top layer is responsible for sharing the statistical strength. A more special case is considered in [88], where only a single example from a new category is provided; thus, it is more difficult to estimate the variance and similarity metric for categorizing an object in this case. It is possible with this model to encode priors for new classes into super-categories. Following the inference of the sub-category to which the novel category belongs, the model can estimate not only the mean of the new category but also an appropriate similarity metric based on parameters inherited from the super-category.

5. Transfer learning using fuzzy system and genetic algorithm

Imprecision, approximation, vagueness and ambiguity of information are driven by the variability encountered when trying to learn an activity with little information. There is a clear co-dependency on the level of certainty in any learning activity and the amount of information that is available, and problems with little information, can have a high degree of uncertainty.

This is why couple of researches appears very recently to apply fuzzy techniques into transfer learning. The use of fuzzy logic allows for the incorporation of approximation and a greater expressiveness of the uncertainty within the knowledge transfer. Zadeh [89] introduced the concept of fuzzy sets which he later expanded on by introducing further aspects of Fuzzy Logic, including fuzzy rules in [90]. The two primary elements within fuzzy logic, the linguistic variable and the fuzzy if-then rule, are able to mimic the human ability to capture imprecision and uncertainty within knowledge transfer. Fuzzy logic forms a major component of the published Fuzzy Transfer Learning techniques. Behbood et al. [91, 92] developed a fuzzy-based transductive transfer learning for long term bank failure prediction in which the distribution of data in the source domain differs from that in the target domain. They applied three classical predictors, Neural Network, Support Vector Machine and Fuzzy Neural Network, to predict the initial labels for samples in the target domain, then attempted to refine the labels using fuzzy similarity measures. The authors subsequently improved the performance of the fuzzy refinement domain adaptation method [93] by developing a novel fuzzy measure to simultaneously take account of the similarity and dissimilarity in the refinement process. The proposed method has been applied to text categorization and bank failure prediction. The experimental results demonstrated the superior performance of the proposed method compared to popular classical transductive transfer learning methods. Using fuzzy techniques in similarity measurement and label production, the authors revealed the advantage of fuzzy logic in knowledge transfer where the target domain lacks critical information and involves uncertainty and vagueness. Shell and Coupland domains [94, 95] proposed a framework of fuzzy transfer learning to form a prediction model in intelligent environments. To address the issues of modeling environments in the presence of uncertainty and noise, they introduced a fuzzy logic-based transfer learning that enables the absorption of the inherent uncertainty and dynamic nature of transfer knowledge in intelligent environments. They created a transferable fuzzy inference system using labeled data in the source domain, then adapted and applied the resultant rule base to predict the labels for samples in the target domain. The source rules were adjusted and adapted to the target domain using the Euclidean distance measure. The proposed method was examined in two simulated intelligent environments. The experimental results demonstrated the superior performance of fuzzy transfer learning compared to classical prediction models; however the method has not been compared with transfer learning methods. Deng et al. [96] proposed the generalized hidden-mapping ridge regression (GHRR) method to train various types of classical intelligence models, including neural networks, fuzzy logic systems and kernel methods. The knowledge-leverage based transfer learning mechanism is integrated with GHRR to realize the inductive transfer learning method called transfer GHRR (TGHRR). Since the information from the induced knowledge is much clearer and more concise than the information from the data in the source domain, it is more convenient to control and balance the similarity and difference of data distributions between the source and target domains. The proposed GHRR and TGHRR algorithms have been evaluated experimentally by performing regression and classification on synthetic and real world datasets. The results demonstrated that the performance of TGHRR is competitive with or even superior to existing state-of-the-art inductive transfer learning algorithms.

Genetic algorithm is an evolutionary method that simulates the process of natural selection to solve mainly optimization and search problems. This method uses techniques inspired by natural evolution such as inheritance, mutation, selection and crossover. Koçer and Arslan [97] introduced the use of genetic algorithm and transfer learning by extending a previously constructed algorithm. Their approach was to extend the transfer learning method of producing a translation function. This process allows for differing value functions that have been learnt to be mapped from source to target tasks. The authors incorporated the use of a set of policies originally constructed by a genetic algorithm to form the initial population for training the target task. They showed that the transfer of inter-task mappings can reduce the time required to learn a second, more complex task.

6. Applications of transfer learning

Transfer learning approaches with the support of computational intelligence methods such as neural network, Bayesian network, and fuzzy logic have been applied in real-world applications. These applications largely fall into the following five categories: 1) Nature language processing; 2) Computer vision; 3) Biology; 4) Finance; and 5) Business management.

6.1. Nature language processing

Nature language processing (NLP), which can be regarded as the study of human languages, is proposed to make natural language processing interpretable by computers. In general, there are numerous sub-learning tasks in NLP fields, such as text-based learning problems (e.g., text classification or non-topical text analysis), language knowledge understanding, etc.

For text related analysis, i.e., exploring the useful information from a given document, the learning problem of how to label the text documents across different distributions was addressed [17]. In this setting, the labeled training samples shared different distributions from the unlabeled test data. Accordingly, a novel transfer-learning algorithm based on an Expectation-Maximization based Naive Bayes model was proposed for further learning, which has demonstrated the best performance on three different types of data sets. Moreover, considering that most existing transfer learning methods assume that features and labels are numeric, and lack the ability to handle the uncertainty property, Behbood et al. [98] proposed a Fuzzy Domain Adaptation (FDA) approach and carried out an investigation of its applicability to text classification. In addition, for sentiment classification, which is a key challenge in non-topical text analysis, transfer learning technique is also applicable, such as adapting naïve Bayes to domain adaptation for sentiment analysis by fully utilizing the information from both the old-domain and unlabeled new-domain data sets [16].

Furthermore, the transfer learning approach can be used to deal with language knowledge understanding problems. For speech recognition, for example, Swietojanski et al. [52] exploited untranscribed acoustic data to the target languages in a deep neural network on unsupervised cross-lingual knowledge transfer. Similarly, Huang et al. [47] dealt with the cross-language knowledge transfer learning tasks by a shared-hidden-layer multi-lingual deep neural network.

6.2. Computer vision and image processing

The computer vision applied to transfer learning using computational intelligence includes methods for acquiring, processing, analysing, and understanding images, especially in high-dimensional data from the real world, for producing numerical or symbolic information. In this section, we summarize computer visual applications for camera images processing, from digits to letters processing, and video processing.

In early camera image applications based on computational intelligent transfer learning, all approaches used a database of camera images of different objects, each of which had a distinct color or size and was used for vision learning, such as ALVINN-like road-following vision recognition [54]. One challenge in image object recognition is that the distributions of the training images and test images are different. Thus, Chopra et al. [51] argued that in the representation learning camp for images, existing deep learning approaches could not encode the distributional shift between the source and target domains. To this end, the authors proposed a novel transfer deep learning method for object recognition which allows the application of deep learning for domain adaptation. Camera images were also used to solve robotics problems. A visual object tracking routine, which recognizes and tracks the marker in real time, challenged robot researchers [99, 100] that robot-mounted camera [54] was employed for task mappings, e.g. RoboCup soccer Keepaway [101]. Recently, image learning has mainly been used for human facial recognition, e.g., gender and ethnicity recognition based on facial appearance [46], emotional facial appearance recognition derived from synthetic images of neutral faces to that of corresponding images of angry, happy and sad faces [60], age estimations from face images [65], and gaze gesture recognition by eye tracking devices and eye gaze technologies [94]. Knowledge transfer between different handwritten character recognition tasks [48] is another kind of application of transfer learning in computer vision. Kandaswamy et al. [50] trained a neural network to classify Latin digits (specific source problem) and reused it to classify a lowercase letters (different but related target problem) without having to train it from scratch. In the empirical analysis, the authors used the proposed neural network to transfer knowledge from Arabic digits to Latin digits as well. Authors [50] also considered a problem of classifying images of English lowercase a-to-z by reusing fine-tuned features of English handwritten digits 0-to-9.

By applying salient feature detection and tracking in videos to simulate fixations and smooth pursuit in human vision, Zou et al. [102] successfully implemented an unsupervised learning algorithm in a self-taught learning setting. With concrete recognition, features learned from natural videos do not only apply to still images, but also give competitive results on a number of object recognition benchmarks.

6.3. Biology

Transfer learning has been applied to biology fields, including medical problems, biological modeling designs, ecology issues, and so on. In applications related to medical issues, Caruana [54] suggested using multi-task learning in artificial neural networks, and proposed an inductive transfer learning approach for pneumonia risk prediction. A life-long inductive learning approach [56] retained task knowledge in a representational form and transferred knowledge in another form of virtual examples on three heart disease domains, through a neural network-based multi-task learning algorithm. They also put forward another type of sequential inductive transfer model for a medical diagnostics task, i.e., coronary artery disease diagnosis [57]. Recently, Oyen and Lane [79] argued that existing transfer learning methods for Bayesian networks focus on a single posteriori estimation, and that in doing so, other models may be ignored. To this end, they proposed a transfer multi-Bayesian Networks model for whole-brain neuroimaging.

From the aspect of biological modeling designs, e.g., robot bionics, Celiberto Jr et al. [66] combined three artificial intelligence techniques, case-based reasoning, heuristically accelerated reinforcement learning and neural networks, in a transfer learning problem. They then proposed a novel model called L3 to speed up the reinforcement learning framework for a set of empirical evaluations between two domains (the Acrobot and the Robocup 3D). Another important biology domain, ecology, has attracted the attention of researchers into transfer learning. For instance, Niculescu-Mizil and Caruana [75] proposed a multi-task Bayesian network structure learning (i.e., inductive transfer) to re-evaluate the performance of ALARM (a logical alarm reduction mechanism) and to handle a real bird ecology problem in North America.

6.4. Finance

Another application area of transfer learning is finance, such as in the area of car insurance risk estimations and financial early warning systems. Niculescu-Mizil and Caruana [75] presented an inductive transfer learning approach, which jointly learns multiple Bayesian network structures instead of adaptive probabilistic networks from multiple related data sets. The authors examined the proposed method using car insurance risk estimation networks. It is worth noticing that the works on intelligent financial warning systems and long term prediction in banking ecosystems [91-93] are the first systematic studies to apply transfer learning approaches using fuzzy logic techniques of computational intelligence to real-world financial applications to exploit the knowledge of the banking system, e.g., transferring the information from one country to establish a prediction model in another country.

6.5. Business management

Transfer learning using computational intelligence has been applied in business management. For instance, Roy and Kaelbling [71] proposed an efficient Bayesian task-level transfer learning to tackle the user's behavior in the meeting domain. Jin and Sun [103] indicated that traditional neural network methods for traffic flow forecasting are based on a single task which cannot utilize information from other tasks. To address this challenge, multi-task based neural network is proposed to transfer knowledge to deal with traffic flow forecasting. Luis, Sucar and Morales [76] proposed the use of a novel transfer Bayesian network learning framework, including structure and parameter learning, to handle a product manufacture process issue. Recently, Ma et al. [72] studied the cross-company software defect prediction scenario in which the source and target data sets come from different companies, and proposed a novel transfer naive Bayes as the solution. A dynamic model for intelligent environments has been proposed to make use of the data from different feature spaces and domains [94, 104], with a novel fuzzy transfer learning process.

7. Comprehensive analysis and findings

In this paper, we have reviewed current trends of computational intelligence-based transfer learning and their applications. According to the review, computational intelligence techniques used in transfer learning can be classified into three main groups: Neural Network, Bayes and Fuzzy Logic, and Genetic Algorithm. The number of reviewed transfer learning papers for each computational intelligence technique in each application domain is summarized and presented in Table 1. From the summary of transfer learning, it is concluded that transfer learning with the use computational intelligence, as an emerging research topic, starts playing an important role in almost all kinds of application. Of the computational intelligence methods, neural network has been extensively used for transfer learning, mainly in computer vision and image processing domain. The main reason why neural network has been widely used in transfer learning is that it doesn't have I.I.D. assumption on data while almost all stochastic techniques have. It can also be identified that Fuzzy-based transfer learning techniques have played an increasingly important role in recent applications particularly finance. Since many real world applications have noisy and uncertainty in data, researchers take fuzzy systems into account for transfer learning more and more.

Table 1. Summary of transfer learning techniques in each application domain

Techniques Domains	Artificial Neural Networks	Bayes	Fuzzy logic	No. of listed references
Natural language processing	2	2	1	5
Computer vision & image processing	11	0	1	12
Biology	4	2	0	6
Finance	0	1	3	4
Business management	1	3	2	6
Total	**17**	**7**	**6**	**30**

In the future, several important research challenges in the field of computational intelligence-based transfer learning need to be addressed. First, the computational complexity is a crucial issue in computational intelligence-based transfer learning. Almost, all reviewed studies have focused on accuracy as a measurement for model performance. However, comparing with the statistical transfer learning methods, computational intelligence techniques usually gain more computational complexity which should be handled. In addition, how to avoid negative transfer is an open problem which is not only in the classical transfer learning but also in computational intelligence-based transfer learning. The transferability among source and target domains needs to be studied profoundly and a comprehensive and accurate transferability measures to be implemented that can guarantee the negative learning will not happens. Moreover, almost all reviewed studies have assumed that the feature spaces between the source and target domains are the same. However, in many applications, which we wish to transfer knowledge among domains, this assumption cannot be held. This type of transfer learning which is referred as the heterogeneous transfer learning has not been addressed in computational intelligence-based transfer learning literature. Finally, so far the computational intelligence techniques are applied for small scale transfer learning problems. Nonetheless, in the era of big data, there are many interesting applications such as social network analysis and web-based recommender systems that can exploit transfer learning and computational intelligence techniques. The capability of computational intelligence to handle non-I.I.D. noisy data can pave the way to use these techniques in big scale real world applications.

Two important features of this paper clearly distinguish it from other survey papers in the transfer learning area: Firstly, It targets the development of transfer learning methods that use computational intelligence. Secondly, It systematically examines the applications of transfer learning that are integrated with computational intelligence.

We believe that this paper can provide researchers and practical professionals with state-of-the-art knowledge on transfer learning with computational intelligence and give guidelines about how to develop and apply transfer learning in different domains to support users in various decision activities.

Acknowledgment

The work presented in this paper was supported by the Australian Research Council (ARC) under discovery grant DP140101366.

References

1. Zhu, X., *Semi-Supervised Learning Literature Survey*. 2005, University of Wisconsin-Madison: Madison, WI.
2. Nigam, K., et al., *Text classification from labeled and unlabeled documents using EM.* Machine Learning, 2000. **39**(2-3): p. 103-134.
3. Blum, A. and T. Mitchell. *Combining labeled and unlabeled data with co-training*. in *Eleventh Annual Conference on Computational Learning Theory*. 1998. Madison, WI.
4. Joachims, T. *Transductive inference for text classification using support vector machines*. in *Sixteenth International Conference on Machine Learning*. 1999. Bled, Slovenia.
5. Fung, G.P.C., et al., *Text classification without negative examples revisit.* IEEE Transactions on Knowledge and Data Engineering, 2006. **18**(1): p. 6-20.
6. Yin, X., et al., *Efficient classification across multiple database relations: A CrossMine approach.* IEEE Transactions on Knowledge and Data Engineering, 2006. **18**(6): p. 770-783.
7. Kuncheva, L.I. and J.J. Rodriguez, *Classifier ensembles with a random linear oracle.* IEEE Transactions on Knowledge and Data Engineering, 2007. **19**(4): p. 500-508.
8. Baralis, E., S. Chiusano, and P. Garza, *A lazy approach to associative classification.* IEEE Transactions on Knowledge and Data Engineering, 2008. **20**(2): p. 156-171.
9. Pan, S.J. and Q. Yang, *A survey on transfer learning.* IEEE Transactions on Knowledge and Data Engineering, 2010. **22**(10): p. 1345-1359.

10. Klinkenberg, R. and T. Joachims. *Detecting concept drift with support vector machines*. in *Seventeenth International Conference on Machine Learning*. 2000. Stanford, CA.

11. Margolis, A., *A Literature Review of Domain Adaptation with Unlabeled Data*. 2011, University of Washington: Washington.

12. Heckman, J.J., *Sample Selection Bias as a Specification Error.* Econometrica, 1979. **47**(1): p. 61.

13. Huang, J., et al., *Correcting sample selection bias by unlabeled data.* Advances in Neural Information Processing Systems, 2007. **19**: p. 601-608.

14. Sugiyama, M., et al., *Direct importance estimation with model selection and its application to covariate shift adaptation.* Advances in Neural Information Processing Systems, 2008. **20**: p. 1433-1440.

15. Tsuboi, Y., et al., *Direct density ratio estimation for large-scale covariate shift adaptation.* Information and Media Technologies, 2009. **4**(2): p. 529-546.

16. Tan, S., et al. *Adapting naive Bayes to domain adaptation for sentiment analysis*. in *31st European Conference on Advances in Information Retrieval*. 2009. Toulouse, France.

17. Dai, W., et al. *Transferring naive Bayes classifiers for text classification*. in *22nd National Conference on Artificial Intelligence*. 2007. Vancouver, Canada.

18. McClosky, D., E. Charniak, and M. Johnson. *Reranking and self-training for parser adaptation*. in *21st International Conference on Computational Linguistics*. 2006. Sydney, Australia.

19. Roark, B. and M. Bacchiani. *Supervised and unsupervised PCFG adaptation to novel domains*. in *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language*. 2003. Edmonton, Canada.

20. Sagae, K. and J. Tsujii. *Dependency parsing and domain adaptation with LR models and parser ensembles*. in *11th Conference on Computational Natural Language Learning*. 2007. Prague, Czech Republic.

21. Sagae, K. *Self-training without reranking for parser domain adaptation and its impact on semantic role labeling*. in *Workshop on Domain Adaptation for Natural Language Processing*. 2010. Uppsala, Sweden.

22. Jiang, J. and C.X. Zhai. *Instance weighting for domain adaptation in NLP*. in *45th Annual Meeting of the Association of Computational Linguistics*. 2007. Prague, Czech Republic.

23. Sandu, O., et al. *Domain adaptation to summarize human conversations*. in *Workshop on Domain Adaptation for Natural Language Processing*. 2010. Uppsala, Sweden.

24. Jiang, J. and C.X. Zhai. *A two-stage approach to domain adaptation for statistical classifiers*. in *Sixteenth ACM Conference on Information and Knowledge Management*. 2007. Lisbon, Portugal.

25. Ciaramita, M. and O. Chapelle. *Adaptive parameters for entity recognition with perceptron HMMs*. in *Workshop on Domain Adaptation for Natural Language Processing*. 2010. Uppsala, Sweden.

26. Tan, S., et al. *Using unlabeled data to handle domain-transfer problem of semantic detection*. in *ACM Symposium on Applied Computing*. 2008. Fortaleza, Brazil.

27. Rigutini, L., M. Maggini, and B. Liu. *An EM based training algorithm for cross-language text categorization*. in *IEEE/WIC/ACM International Conference on Web Intelligence*. 2005. Compiegne, France.

28. Shi, L., R. Mihalcea, and M. Tian. *Cross language text classification by model translation and semi-supervised learning*. in *Conference on Empirical Methods in Natural Language Processing*. 2010. Cambridge, MA.

29. Jeong, M., C.Y. Lin, and G.G. Lee. *Semi-supervised speech act recognition in emails and forums*. in *Conference on Empirical Methods in Natural Language Processing*. 2009. Singapore, Singapore.

30. Arnold, A., R. Nallapati, and W.W. Cohen. *A Comparative Study of Methods for Transductive Transfer Learning*. in *Seventh IEEE International Conference on Data Mining Workshops*. 2007. Omaha, NE.

31. Aue, A. and M. Gamon. *Customizing sentiment classifiers to new domains: A case study*. in *International Conference on Recent Advances in Natural Language Processing*. 2005. Borovets, Bulgaria.

32. Satpal, S. and S. Sarawagi. *Domain adaptation of conditional probability models via feature subsetting*. in *11th European Conference on Principles and Practice of Knowledge Discovery in Databases*. 2007. Warsaw, Poland.

33. Chen, B., et al. *Extracting discriminative concepts for domain adaptation in text mining*. in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2009. Paris, France.

34. Pan, S.J., J.T. Kwok, and Q. Yang. *Transfer learning via dimensionality reduction*. in *23rd National Conference on Artificial Intelligence*. 2008. Chicago, IL.

35. Pan, S.J., et al., *Domain adaptation via transfer component analysis.* IEEE Transactions on Neural Networks, 2011. **22**(2): p. 199-210.

36. Blitzer, J., R. McOnald, and F. Pereira. *Domain adaptation with structural correspondence learning*. in *Conference on Empirical Methods in Natural Language Processing*. 2006. Sydney, Australia.

37. Blitzer, J., et al. *Learning bounds for domain adaptation*. in *Twenty-First Annual Conference on Neural Information Processing Systems*. 2007. Cambridge, MA.

38. Ben-David, S., et al., *A theory of learning from different domains.* Machine Learning, 2010. **79**(1): p. 151-175.

39. Huang, F. and A. Yates. *Exploring representation-learning approaches to domain adaptation*. in *Workshop on Domain Adaptation for Natural Language Processing*. 2010. Uppsala, Sweden.

40. Huang, F. and A. Yates. *Open-domain semantic role labeling by modeling word spans*. in *48th Annual Meeting of the Association for Computational Linguistics*. 2010. Uppsala, Sweden.

41. Huang, F. and A. Yates. *Distributional representations for handling sparsity in supervised sequence-labeling*. in *Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Conference on Natural Language Processing*. 2009. Singapore, Singapore.

42. Pan, S.J., et al. *Cross-domain sentiment classification via spectral feature alignment*. in *19th International Conference on World Wide Web*. 2010. Raleigh, NC.

43. Gao, J., et al. *Knowledge transfer via multiple model local structure mapping*. in *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2008. Las Vegas, NV.

44. Xue, G.-R., et al. *Topic-bridged PLSA for cross-domain text classification*. in *31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2008. Singapore, Singapore.

45. Hubel, D.H. and T.N. Wiesel, *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.* Journal of Physiology, 1962. **160**(1): p. 106.

46. Ahmed, A., et al., *Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks*, in *Computer Vision–ECCV 2008*. 2008, Springer. p. 69-82.

47. Huang, J.-T., et al. *Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers*. in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013. IEEE.

48. Ciresan, D.C., U. Meier, and J. Schmidhuber. *Transfer learning for Latin and Chinese characters with deep neural networks*. in *International Joint Conference on Neural Networks (IJCNN)*. 2012. IEEE.

49. Collobert, R. and J. Weston. *A unified architecture for natural language processing: Deep neural networks with multitask learning*. in *25th international conference on Machine learning*. 2008. ACM.

50. Kandaswamy, C., et al. *Improving deep neural network performance by reusing features trained with transductive transference*. in *Proceedings of the 24th International Conference on Artificial Neural Networks*. 2014.

51. Chopra, S., S. Balakrishnan, and R. Gopalan. *DLID: Deep learning for domain adaptation by interpolating between domains*. in *ICML Workshop on Challenges in Representation Learning*. 2013.

52. Swietojanski, P., A. Ghoshal, and S. Renals. *Unsupervised cross-lingual knowledge transfer in DNN-based LVCSR*. in *IEEE Workshop on Spoken Language Technology (SLT)*. 2012.
53. Caruana, R. *Multitask learning: A knowledge-based source of inductive bias*. in *Tenth International Conference of Machine Learning*. 1993.
54. Caruana, R., *Multitask learning.* Machine Learning, 1997. **28**: p. 41-75.
55. Silver, D. and R. Mercer. *Selective functional transfer: Inductive bias from related tasks*. in *International Conference on Artificial Intelligence and Soft Computing (ASC)*. 2001.
56. Silver, D.L. and R.E. Mercer, *The task rehearsal method of life-long learning: Overcoming impoverished data*, in *Advances in Artificial Intelligence*. 2002, Springer. p. 90-101.
57. Silver, D.L. and R.E. Mercer. *Sequential inductive transfer for coronary artery disease diagnosis*. in *International Joint Conference on Neural Networks (IJCNN)* 2007. IEEE.
58. Silver, D.L. and R. Poirier. *Context-Sensitive MTL Networks for Machine Lifelong Learning*. in *20th Florida Artificial Intelligence Research Society (FLAIRS) Conference*. 2007.
59. Silver, D.L., R. Poirier, and D. Currie, *Inductive transfer with context-sensitive neural networks.* Machine Learning, 2008. **73**(3): p. 313-336.
60. Silver, D.L. and L. Tu, *Image transformation: Inductive transfer between multiple tasks having multiple outputs*, in *Advances in Artificial Intelligence*. 2008, Springer. p. 296-307.
61. Yamauchi, K., *Covariate shift and incremental learning*, in *Advances in Neuro-Information Processing*. 2009, Springer. p. 1154-1162.
62. Shimodaira, H., *Improving predictive inference under covariate shift by weighting the log-likelihood function.* Journal of Statistical Planning and Inference, 2000. **90**(2): p. 227-244.
63. Yamauchi, K., *Optimal incremental learning under covariate shift.* Memetic Computing, 2009. **1**(4): p. 271-279.
64. Liu, W., H. Zhang, and J. Li. *Inductive transfer through neural network error and dataset regrouping*. in *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)* 2009. IEEE.
65. Ueki, K., M. Sugiyama, and Y. Ihara. *Perceived age estimation under lighting condition change by covariate shift adaptation*. in *20th International Conference on Pattern Recognition (ICPR)* 2010. IEEE.
66. Celiberto Jr, L.A., et al. *Using cases as heuristics in reinforcement learning: A transfer learning application*. in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. 2011.
67. Lewis, D.D., *Representation and learning in information retrieval*. 1992, University of Massachusetts.
68. Kononenko, I., *Inductive and Bayesian learning in medical diagnosis.* Applied Artificial Intelligence an International Journal, 1993. **7**(4): p. 317-337.
69. Androutsopoulos, I., et al. *An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages*. in *23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2000. ACM.
70. Sebastiani, F., *Machine learning in automated text categorization.* ACM Computing Surveys (CSUR), 2002. **34**(1): p. 1-47.
71. Roy, D.M. and L.P. Kaelbling. *Efficient Bayesian task-level transfer learning*. in *International Joint Conference on Artificial Intelligence*. 2007.
72. Ma, Y., et al., *Transfer learning for cross-company software defect prediction.* Information and Software Technology, 2012. **54**(3): p. 248-256.
73. Heckerman, D., *A tutorial on learning with Bayesian networks*. 1998: Springer.
74. Buntine, W.L., *A guide to the literature on learning probabilistic networks from data.* Knowledge and Data

Engineering, IEEE Transactions on, 1996. **8**(2): p. 195-210.

75. Niculescu-Mizil, A. and R. Caruana. *Inductive transfer for Bayesian network structure learning*. in *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2007.

76. Luis, R., L.E. Sucar, and E.F. Morales, *Inductive transfer for learning Bayesian networks.* Machine Learning, 2010. **79**(1-2): p. 227-255.

77. Richardson, M. and P. Domingos. *Learning with knowledge from multiple experts*. in *International Conference on Machine learning (ICML)*. 2003.

78. Oyen, D. and T. Lane. *Leveraging domain knowledge in multitask Bayesian network structure learning*. in *26th Association for the Advancement of Artificial Intelligence (AAAI) Conference*. 2012.

79. Oyen, D. and T. Lane. *Bayesian discovery of multiple Bayesian networks via transfer learning*. in *13th IEEE International Conference on Data Mining (ICDM)*. 2013. IEEE.

80. Friedman, N. and D. Koller, *Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks.* Machine Learning, 2003. **50**(1-2): p. 95-125.

81. Koivisto, M. and K. Sood, *Exact Bayesian structure discovery in Bayesian networks.* The Journal of Machine Learning Research, 2004. **5**: p. 549-573.

82. Wilson, A., et al. *Multi-task reinforcement learning: A hierarchical Bayesian approach*. in *24th International Conference on Machine Learning*. 2007. ACM.

83. Wilson, A., A. Fern, and P. Tadepalli. *Transfer Learning in sequential decision problems: A hierarchical Bayesian approach*. in *International Conference of Machine Learning (Unsupervised and Transfer Learning)*. 2012.

84. Yang, P., Q. Tan, and Y. Ding. *Bayesian task-level transfer learning for non-linear regression*. in *International Conference on Computer Science and Software Engineering*. 2008. IEEE.

85. Raykar, V.C., et al. *Bayesian multiple instance learning: Automatic feature selection and inductive transfer*. in *25th International Conference on Machine Learning*. 2008. ACM.

86. Finkel, J.R. and C.D. Manning. *Hierarchical Bayesian domain adaptation*. in *Annual Conference of the North American Chapter of the Association for Computational Linguistics*. 2009. Association for Computational Linguistics.

87. Wood, F. and Y.W. Teh. *A hierarchical nonparametric Bayesian approach to statistical language model domain adaptation*. in *International Conference on Artificial Intelligence and Statistics*. 2009.

88. Salakhutdinov, R., J. Tenenbaum, and A. Torralba, *One-shot learning with a hierarchical nonparametric Bayesian model*, MIT-CSAIL-TR-2010-052, Editor. 2010, MIT.

89. Zadeh, L.A., *Fuzzy sets.* Information and Control, 1965. **8**(3): p. 338-353.

90. Bellman, R.E. and L.A. Zadeh, *Decision-making in a fuzzy environment.* Management Science, 1970. **17**(4): p. 141-164.

91. Behbood, V., L. Jie, and Z. Guangquan. *Long term bank failure prediction using fuzzy refinement-based transductive transfer learning*. in *IEEE International Conference on Fuzzy Systems (FUZZ)*. 2011.

92. Behbood, V., J. Lu, and G. Zhang, *Fuzzy bridged refinement domain adaptation: Long-term bank failure prediction.* International Journal of Computational Intelligence and Applications, 2013. **12**(01).

93. Behbood, V., L. Jie, and Z. Guangquan, *Fuzzy refinement domain adaptation for long term prediction in banking ecosystem.* IEEE Transactions on Industrial Informatics, 2014. **10**(2): p. 1637-1646.

94. Shell, J. and S. Coupland, *Towards fuzzy transfer learning for intelligent environments*, in *Ambient Intelligence*, F. Paternò, et al., Editors. 2012, Springer Berlin Heidelberg. p. 145-160.

95. Shell, J. and S. Coupland, *Fuzzy transfer learning: Methodology and application.* Information Sciences, 2015.

293(0): p. 59-79.

96. Deng, Z., et al., *Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel method.* IEEE Transactions on Cybernetics, 2014. **44**(12): p. 2585-2599.
97. Koçer, B. and A. Arslan, *Genetic transfer learning.* Expert Systems with Applications, 2010. **37**(10): p. 6997-7002.
98. Behbood, V., L. Jie, and Z. Guangquan. *Text categorization by fuzzy domain adaptation*. in *IEEE International Conference on Fuzzy Systems*. 2013.
99. Thrun, S., *Is learning the n-th thing any easier than learning the first?* Advances in Neural Information Processing Systems, 1996: p. 640-646.
100. Thrun, S. *A lifelong learning perspective for mobile robot control*. in *IEEE/RSJ/GI International Conference on Intelligent Robots and Systems* 1994. IEEE.
101. Taylor, M.E., P. Stone, and Y. Liu, *Transfer Learning via Inter-Task Mappings for Temporal Difference Learning.* Journal of Machine Learning Research, 2007. **8**(1): p. 2125-2167.
102. Zou, W., et al. *Deep learning of invariant features via simulated fixations in video*. in *Advances in Neural Information Processing Systems*. 2012.
103. Jin, F. and S. Sun. *Neural network multitask learning for traffic flow forecasting*. in *IEEE International Joint Conference on Neural Networks (IJCNN)*. 2008. IEEE.
104. Shell, J., *Fuzzy transfer learning*. 2013, De Montfort University.