Author Name

Book title goes here

Symbol Description

α	To solve the generator maintenance scheduling, in the past, several mathematical techniques have been applied.
σ^2	These include integer programming, integer linear programming, dynamic programming, branch and bound etc.
$\sum$	Several heuristic search algorithms have also been developed. In recent years expert systems,
abc	fuzzy approaches, simulated annealing and genetic algorithms have also been tested.
$\theta\sqrt{abc}$	This paper presents a survey of the literature
ζ	over the past fifteen years in the generator
∂	maintenance scheduling. The objective is to
sdf	present a clear picture of the available recent literature
ewq	of the problem, the constraints and the other aspects of
bvcn	the generator maintenance schedule.

Chapter 1

Transfer Learning, Multi-task Learning, and Cost-sensitive Learning

1.1 Introduction

In this chapter we discuss cost sensitive learning in the context of transfer learning and multi-task learning problems. In many machine learning problems, the learning problem in one or more target domains may be very difficult to solve, but we may have some related knowledge from one or more different but similar domains. In such cases, we may find some common knowledge between these domains to help improve the learning performance in some chosen target domains, or improve the performance of learning in all related domains. Learning under these circumstances is called transfer learning or multi-task learning (see a survey by Pan and Yang [42]).

This learning paradigm has been inspired by human learning activities in that people often apply the knowledge gained from previous learning tasks to help learn a new task. For example, a baby can be observed to first learn how to recognize its parents before using this knowledge to help it learn how to recognize other people. Multi-task learning can be formulated under two different settings: *symmetric* and *asymmetric* [61]. While symmetric multi-task learning seeks to improve the performance of all tasks simultaneously, the objective of

TABLE 1.1: Notations

$\mathcal{D}$	A dataset
$\mathcal{X}$	A feature space
$\mathcal{H}$	A hypothesis space
$\text{tr}(\mathbf{A})$	Trace of matrix $\mathbf{A}$
p	A probability distribution
$\mathbb{E}_p[\cdot]$	Expectation with respect to distribution P
$\mathcal{S}$	The source task (domain)
$\mathcal{T}$	The target task (domain)
p_s	Distribution on the training/source data
p_t	Distribution on test/target data
$\mathbb{E}_{\mathcal{D}}[\cdot]$	Expectation over dataset $\mathcal{D}$
min	Minimize
max	Maximize

asymmetric multi-task learning is to improve the performance of some target task using information from the source tasks, typically after the source tasks have been learned using some symmetric multi-task learning method. In this sense, asymmetric multi-task learning is related to *transfer learning* [42], but the major difference is that the source tasks are still learned simultaneously in asymmetric multi-task learning, while they are learned independently in transfer learning.

When the costs of different losses are considered, transfer and multi-task learning can be further formulated more specifically depending on the different objectives to optimize. In transfer learning, cost-sensitive learning can be seen as the process in which we only consider the cost on the interested target tasks, whereas in multi-task learning, we further consider the costs as evenly distributed over all tasks under consideration. In the following, we first consider the transfer learning model where there is only one target learning task. We then consider the symmetric multi-task model where tasks have equal weights.

1.2 Notations

We first introduce the notations we will use in later sections. In general, we use lower-case letters like a, b, c to represent scalars and bold letter case like $\mathbf{u}, \mathbf{v}, \mathbf{w}$ to represent vectors. Upper-case letter in bold like $\mathbf{A}, \mathbf{B}, \mathbf{C}$ represent matrices. Lower-case letters with parenthesis as $f(), g(), h()$ represent functions. More specific notations are shown in Table 1.1. Other notations will be introduce where they are using.

1.3 Transfer Learning Models

Cost-sensitive learning can be seen as placing a heavier importance emphasis on some selected instances and features than others. In transfer learning, these weight assignments correspond to sampling based approaches in covariate shift. These learning approaches also share many similarities with classical cost-sensitive learning methods.

Transfer learning attempts to learn useful knowledge from a source task and generalize this knowledge in a target task. This learning paradigm breaks a common assumption made by most machine learning methods, which states that the training and test data are drawn from the same feature space and the same distribution. When the distribution changes, most statistical models need to be rebuilt from scratch using newly collected training data. In many real-world applications, it is expensive or even impossible to re-collect the needed training data and rebuild the models. Therefore, we expect to transfer the knowledge from the source tasks to the target task to reduce the effort of labeling. These learning problems include domain adaptation, sample selection bias, covariate shift and self-taught learning [10, 16, 9, 48]. These approaches are similar in their common goal of knowledge reuse, and different in their specific assumptions made in their learning algorithms to handle the knowledge transfer.

In the following, we review several major methodologies that have been developed to solve the transfer learning problem. These methodologies can be classified in three categories: sampling based approaches, representation based approaches and task-regularization based approaches.

1.3.1 Sampling Based Approach

Sampling-based approaches find their roots in statistical sampling methods, where the aim is to draw selected instances from a particular distribution. When directly drawing samples from the distribution is difficult, samples are drawn from some initial distributions and then are adapted to approximate the original distribution. These adaptation algorithms can be utilized to correct the distribution difference for transfer learning.

1.3.1.1 Importance Sampling as Cost-Sensitive Learning

The intuitive idea behind sampling based approaches is the following. Although the source and the target tasks are different, there are certain parts of the data that can still be reused together with a few labeled data in the target task. Most instance-based transfer approaches are motivated by importance sampling. To see how importance sampling methods may help in this setting, we first review the problem of empirical risk minimization (ERM) [59]. In general, we might want to learn the optimal parameters θ^* of the model by

minimizing the expected risk,

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

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

$$\theta^* = \arg\min_{\theta \in \Theta} \ \frac{1}{n} \sum_{i=1}^{n} l(\mathbf{x}_i, y_i, \theta),$$

where n is size of the training data.

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

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

where $(\mathbf{x}, y) \sim p_t$ means that the data $(\mathbf{x}, y)$ samples follow the distribution given by p_t. When no labeled data in the target domain are observed in training data, we have to learn a model from the source domain data instead. If $p_s(\mathbf{x}, y) = p_t(\mathbf{x}, y)$, then we may simply learn the model by solving the following optimization problem for use in the target domain,

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

Otherwise, when $p_s(\mathbf{x}, y) \neq p_t(\mathbf{x}, y)$, we need to modify the above optimization problem to learn a consistent model for the target domain, as follows:

$$\begin{aligned}
\mathbb{E}_{(\mathbf{x},y)\sim p_t}[l(\mathbf{x}, y, \theta)] &= \int p_t(\mathbf{x}, y) \ l(\mathbf{x}, y, \theta) \mathrm{d}\mathbf{x}\mathrm{d}y \\
&= \int \frac{p_t(\mathbf{x}, y)}{p_s(\mathbf{x}, y)} p_s(\mathbf{x}, y) \ l(\mathbf{x}, y, \theta) \mathrm{d}\mathbf{x}\mathrm{d}y \\
&= \mathbb{E}_{(\mathbf{x},y)\sim p_s}\big[\frac{p_t(\mathbf{x}, y)}{p_s(\mathbf{x}, y)} l(\mathbf{x}, y, \theta)\big]
\end{aligned}$$

Therefore, by adding different importance weights to each instance with the corresponding weight $\beta(\mathbf{x}, y) := \frac{p_t(\mathbf{x},y)}{p_s(\mathbf{x},y)}$, we can learn a consistent model for the target domain.

1.3.1.2 Estimating the Importance Weight

The key problem in the sampling based algorithm is how to estimate the importance weights. In the following, we discuss two recently developed methods.

Under the covariate shift setting, we have the assumption that the conditional distribution is invariant, namely, $p_s(y|\mathbf{x}) = p_t(y|\mathbf{x})$. Thus the difference

between $p_s(\mathbf{x}, y)$ and $p_t(\mathbf{x}, y)$ is caused by $p_s(\mathbf{x})$ and $p_t(\mathbf{x})$ and $\frac{p_t(\mathbf{x},y)}{p_s(\mathbf{x},y)} = \frac{p_s(\mathbf{x})}{p_t(\mathbf{x})}$. If we can estimate $\frac{p(\mathbf{x}_s)}{p(\mathbf{x}_t)}$ for each instance, we can solve the *covariate shift* problems via cost-sensitive learning.

There are various ways to estimate $\frac{p_t(\mathbf{x})}{p_s(\mathbf{x})}$. The most intuitive idea is to estimate p_t and p_s first and then compute the ratio. This basic solution is used in early works for solving the sample selection bias problem [67]. However, this method needs to estimate the density function of distributions, which can be infeasible in high dimensional spaces. Therefore, directly estimating the importance weight $\frac{p_t(\mathbf{x})}{p_s(\mathbf{x})}$ is a more preferable approach [58]. The idea is to estimate the weighting function $\beta(\mathbf{x})$ that can approximate the ratio. Formally, we would want to minimize the following objective function

$$\min \quad \mathrm{div}\big(p_t(\mathbf{x}),\ \beta(\mathbf{x}) \cdot p_s(\mathbf{x})\big) \tag{1.1}$$

where $\mathrm{div}(,)$ is a type of divergence on distributions.

Sugiyama *et al.* propose an algorithm known as Kullback-Leibler Importance Estimation Procedure (KLIEP), which uses the Kullback-Leibler divergence as the objective function [52]. The objective then becomes

$$\min \quad \mathrm{D_{KL}}\big(p_t(\mathbf{x}) \ ||\ \beta(\mathbf{x}) \cdot p_s(\mathbf{x})\big).$$

According to the definition of Kullback-Leibler divergence,

$$\begin{aligned} \mathrm{D_{KL}}\big(p_t(\mathbf{x}) \ ||\ w(\mathbf{x}) \cdot p_s(\mathbf{x})\big) &= \int p_t(\mathbf{x}) \log \frac{p_t(\mathbf{x})}{p_s(\mathbf{x})\beta(\mathbf{x})} \mathrm{d}\mathbf{x} \\ &= \int p_t(\mathbf{x}) \log \frac{p_t(\mathbf{x})}{p_s(\mathbf{x})} \mathrm{d}\mathbf{x} - \int p_t(\mathbf{x}) \log \beta(\mathbf{x}) \mathrm{d}\mathbf{x}. \end{aligned}$$

In the above equations, since the first term is independent of $\beta(\mathbf{x})$, the optimization problem can be converted to a maximization problem for the second term:

$$\max \quad \int p_t(\mathbf{x}) \log \beta(\mathbf{x}) \mathrm{d}\mathbf{x}.$$

KLIEP further assumes the weighting function has a special form, such that

$$w(\mathbf{x}) = \sum_l \alpha_l \varphi_l(\mathbf{x}),$$

where α_l are parameters to be learned from the data samples and $\{\varphi_l(\mathbf{x})\}$ are a set of basis functions such that $\varphi_l(\mathbf{x}) \geq 0$ for all $\mathbf{x}$. The optimization problem can be converted to the following convex optimization problem given finite training samples,

$$\begin{aligned} &\max \sum_{\mathbf{x} \in \mathcal{D}_t} \log(\sum_l \alpha_l \varphi_l(\mathbf{x})), \\ &s.t. \sum_{\mathbf{x} \in \mathcal{D}_t} \sum_l \alpha_l \varphi_l(\mathbf{x}) = n_s \text{ and } \alpha_l \geq 0 \end{aligned} \tag{1.2}$$

where n_s is the number of data in the source domain and the constraint comes from $\int p_t(\mathbf{x})\beta(\mathbf{x})\mathrm{d}\mathbf{x} = 1$.

It is possible to consider other types of divergence instead of KL-divergence. Recently, researchers have made the connection between distributions and kernel methods. This connection is based on the finding that there exists a bijection[1] μ between the space of all probability measures and the marginal polytope induced by the feature map $\Phi(\mathbf{x})$ if F is an reproducing-kernel Hilbert space (RKHS) with a universal kernel $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$ [24].

Huang *et al.* [28] propose a kernel-mean matching (KMM) algorithm to learn $\beta(\mathbf{x})$ directly by matching the means between the source domain data and the target domain data in a reproducing-kernel Hilbert space (RKHS).

$$\min_{\beta} \; ||\mu(p_t) - \mathbb{E}_{x\sim p_t}[\beta(\mathbf{x}) \cdot \Phi(\mathbf{x})]||,$$

The additional constraint of the optimization problem is similar to KLIEP, which is $\beta(\mathbf{x}) \geq 0$ and $\mathbb{E}_{x\sim p_s}[\beta(\mathbf{x})] = 1$.

KMM can be rewritten as the following quadratic programming (QP) optimization problem.

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \frac{1}{2}\boldsymbol{\beta}^T \mathbf{K}\boldsymbol{\beta} - \kappa^{\mathrm{T}}\boldsymbol{\beta} \\ s.t. \quad & \beta_i \in [0, B] \;\; and \;\; |\sum_{i=1}^{n_s} \beta_i - n_s| \leq n_s\epsilon \end{aligned}$$

where B is the upper bound for the importance weights. $\mathbf{K} = \begin{bmatrix} \mathbf{K}_{s,s} & \mathbf{K}_{s,t} \\ \mathbf{K}_{t,s} & \mathbf{K}_{t,t} \end{bmatrix}$ and $\mathbf{K}_{t,t}$ are kernel matrices for the source domain data and the target domain data, respectively. $\kappa_i = \frac{n_s}{n_t}\sum_{j=1}^{n_t} k(\mathbf{x}_i, \mathbf{x}_{t_j})$, where $\mathbf{x}_i$ is an instance from either the source or target domain, while $\mathbf{x}_{t_j}$ is an instance from the target domain.

Different from KLIEP, which estimates the weighting function, KMM directly estimates the weight vector $\boldsymbol{\beta}$ of training samples. When doing this, the out-of-sample problem introduces difficulty in using cross validation to set the parameters in the kernel function. This is not a problem of KLIEP, since it can be integrated with cross validation to perform model selection automatically in two steps: (1) estimating the weights of the source domain data; (2) training models on the re-weighted data. Both KMM and KLIEP avoid estimating the distribution of $\mathbf{x}$, which is not a trivial problem to solve when the data dimension is high. It has been shown that KMM is equivalent to an alternative version of KLIEP [58].

It is also possible to combine the estimation of importance weights with the learning problem in a unified framework. Bickel *et al.* [9] derive a kernel-logistic regression classifier based on this idea. Besides sample re-weighting

[1] A bijection is a function f from a set X to a set Y with the property that, for every y in Y, there is exactly one x in X such that $f(x) = y$.

techniques, Dai *et al.* [17] extend a traditional Naive Bayesian classifier for the transductive transfer learning problems, where unlabeled data are available at training time. For more information on importance sampling and re-weighting methods for covariate shift or sample selection bias, readers can refer to a recently published book [47] by Quionero-Candela *et al.*.

The covariate shift assumption does not hold when $p_s(y|\mathbf{x}) \neq p_t(y|\mathbf{x})$. In this case, we can consider label information to improve the estimation of the importance weights. This would require that some labeled data for the target task be available. Similar to the previous case, directly estimating the joint probability is an intuitive method, but it suffers from the data sparseness problem. An alternative approach is to learn the weights through a boosting-style algorithm, as is done in TrAdaBoost [18].

TrAdaBoost [18] is an extension of the Adaboost algorithm to the transfer learning problem. The original Adaboost algorithm sequentially trains some weak learners so that the subsequently built classifiers are tweaked in favor of those instances misclassified by previous classifiers [22]. The cost of misclassification increases in each next round. TrAdaBoost revises the weighting scheme to filter out the training data that are very different from the test data by automatically adjusting the weights of training instances. The algorithm of TrAdaBoost is shown in Algorithm 1.

At each boosting step, TrAdaBoost increases the relative weight of target instances that are misclassified, as is shown in Figure 1.1. When a source instance is misclassified, its weight is decreased, instead of increased as in classical boosting algorithms. In this way, TrAdaBoost makes use of the source instances that are similar to the target data while distancing from those that are dissimilar. Since class labels are taken into consideration, we can use $p(\mathbf{x}, y)$ to represent similarity between sample instances in source and target domains, allowing the final classifier to be more accurate and robust. For a more detailed description, please see [18].

1.3.2 Feature-Representation Based Approach

The feature-representation based approaches to the transfer learning problem aim at finding "good" feature representations to minimize domain divergence and classification or regression model error. Strategies to find "good" feature representations are different for different types of the source task data. If many labeled data in the source domain are available, supervised learning methods can be used to construct a feature representation. This is similar to common feature representation based approach in multi-task learning, to be discussed in Section 1.4.1.

Formally, the loss function of representation based approach can be defined as

$$\min_{\theta} \quad \mathrm{div}\big(p_t(f_\theta(\mathbf{x})),\ p_s(f_\theta(\mathbf{x}))\big).$$

where f_θ is a function to map $\mathbf{x}$ to a new feature representation.

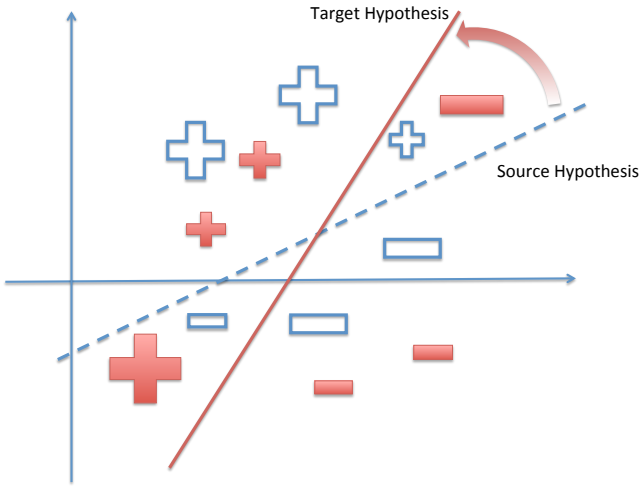

FIGURE 1.1: A toy example to show the weighting scheme for TrAdaBoost. Solid plus and minus points represent the target task and the hollow plus and minus points represent the source task. The symbols +/− indicate positive and negative instances. The size of the points stands for their corresponding weights.

Algorithm 1 TrAdaBoost

Input: Two labeled data sets $\mathcal{D}_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ and $\mathcal{D}_s = \{(\mathbf{x}_j, y_j)\}_{j=n+1}^{n+m}$, the unlabeled data set $\mathcal{D}_u = \{\mathbf{x}_k\}$, a base learning algorithm *Learner*, and the maximum number of iterations T.

Output: A hypothesis $h_f(\mathbf{x})$

Initialize the initial weight vector.

for $t = 1, \ldots, T$ **do**

1. Set $\beta^t \leftarrow \beta^t / (\sum_i \beta_i^t)$

2. Call *Learner*, providing it the combined training set with the weight w^t and the unlabeled data set $\mathcal{D}_u$. Then, get back a hypothesis $h_t : \mathcal{X} \to \{-1, +1\}$.

3. Calculate the error of h_t by

$$\epsilon_t = \sum_{i=n+1}^{n+m} \frac{\beta_i^t \|h_t(\mathbf{x}_i) - y_i\|}{\sum_{i=n+1}^{n+m} \beta_i^t}$$

Set $\alpha_t = \epsilon_t / (1 - \epsilon_t)$ and $\alpha = 1/(1 + \sqrt{2 \ln n / T})$.

Update the new weight vector:

$$\beta_i^{t+1} = \begin{cases} \beta_i^t \alpha^{\|h_t(\mathbf{x}_i) - y_i\|} & \text{if } 1 \le i \le n \\ \beta_i^t \alpha_t^{-\|h_t(\mathbf{x}_i) - y_i\|} & \text{if } n+1 \le i \le n+m \end{cases}$$

and the hypothesis

$$h_f(x) = \begin{cases} 1, & \prod_{t=\lceil T/2 \rceil}^{T} \alpha_t^{-h_t(\mathbf{x})} \ge \prod_{t=\lceil T/2 \rceil}^{T} \alpha_t^{-\frac{1}{2}} \\ 0, & \text{otherwise} \end{cases}$$

end for

This objective is similar to Eq. 1.1 in that they both try to minimize the divergence between transformation of distributions. They differ in that feature-representation based approaches use transformation on features.

We should note that the above objective is only a necessary condition for a successful transfer learning algorithm, because we can always find a trivial mapping function that makes the distributions are identical. Often, we need other constraints or objectives at the same time to avoid such trivial mappings.

If no labeled data are available in the source domain, we can exploit what is known as self-taught learning [48], which is a type of unsupervised learning method that can be used to construct a new feature representation. In self-taught learning, the label space differences between the source and target domains may be large, which implies the auxiliary information of the source domain cannot be used directly. This situation is similar to the inductive transfer-learning setting where the labeled data in the source domain are unavailable. Thus, the key is to find the overlap of the two feature spaces either through a mapping function or through a subspace.

As an example of feature-representation based approach, consider a sentiment classification problem, which aims to find the orientation of product reviews based on their content. For the sentiment classification problem, Blitzer *et al.* [11] propose the structural correspondence learning (SCL) algorithm to exploit domain adaptation techniques for sentiment classification. SCL uses an alternating structural optimization (ASO) algorithm as the optimization algorithm, which was proposed by Ando and Zhang [3]. SCL tries to construct a set of related tasks to model the relationship between pivot features and non-pivot features. The non-pivot features with similar weights among the source and target tasks tend to have similar discriminative power in a low-dimensional latent space, which can be used to transfer the classification knowledge.

Another example is transfer learning via dimensionality reduction, which is proposed by Pan *et al.* [43]. In this work, Pan *et al.* exploit the Maximum Mean Discrepancy Embedding (MMDE) method, originally designed for dimensionality reduction, to learn a low-dimensional space to reduce the difference of distributions between different domains for transfer learning. In particular,

$$\min_{\beta} \ \|\mathbb{E}_{x\sim p_s}[\Phi(\mathbf{x})] - \mathbb{E}_{x\sim p_t}[\Phi(\mathbf{x})]\|,$$

The aim of MMDE is that, besides minimizing the gap between the two distributions, MMDE also maximizes the information to be kept in the kernel space, which is represented by the trace of the kernel matrix.

In the same spirit, Si *et al.* in [51] consider a linear mapping function for f and using the Bregman-divergence as the divergence function, where the objective function can be formulated as follows,

$$\min_{W} \ \mathrm{D}_{\mathrm{Breg}}\big(p_t(f_{\mathbf{W}}(\mathbf{x}))||p_s(f_{\mathbf{W}}(\mathbf{x}))\big) + \lambda \cdot l(\mathbf{W})$$

In this equation, $\mathbf{W}$ is a linear mapping from a high dimensional space to

TABLE 1.2: Cross domain sentiment classification examples

	Electronics	Video Games
P	Compact; easy to operate; very good picture quality; looks sharp!	A very good game! It is action packed and full of excitement. I am very much hooked on this game.
P	I purchased this unit from Circuit City and I was very excited about the quality of the picture. It is really nice and sharp.	Very realistic shooting action and good plots. We played this and were hooked.
N	It is also quite blurry in very dark settings. I will never buy HP again.	The game is so boring. I am extremely unhappy and will probably never buy UbiSoft again.

a low dimensional space and $l(\mathbf{W})$ is a general subspace learning objective function.

The feature-representation based approaches are also used in multi-task learning problem, which we review in Section 1.4.1.

1.3.3 Application Example: Cross-domain Sentiment Classification

We take *Sentiment classification* as an example, which aims to determine whether a product review document reflects a positive or a negative view. Sentiment classification is very useful in online shopping applications, since it allows vendors to know which products are liked by the customers and for what reasons. Many machine learning techniques have been developed for sentiment classification, which have shown good performance when there are sufficient labeled data for training in one specific domain.

Sentiment classification is a domain-dependent task. Table 1.2 gives some examples on domain differences of sentiment terms. Table 1 shows several user review sentences from two domains: electronics and video games. In the electronics domain, we may use words like "compact", "sharp" to express our positive sentiment and use "blurry" to express our negative sentiment. While in the video game domain, words like "hooked", "realistic" indicate positive opinion and the word "boring" indicates negative opinion. Due to the mismatch between domain-specific words, a sentiment classifier trained in one domain may not work well when directly applied to other domains. Thus, cross-domain sentiment classification algorithms are highly desirable for reducing the domain dependency and manually labeling cost.

Among the cross-domain sentiment classification algorithms, most adopt the feature-representation based approaches. As we see in Table 1.2, some sentiment terms are shared across domains while some are domain-dependent.

TABLE 1.3: Results on Sentiment Classification.

Dataset	No Transfer	SCL	SFA	Bound
Book → DVD	77.3%	78.5%	81.4%	82.6%
DVD → Book	74.1%	77.6%	77.1%	81.4%
Kitchen → Electronic	82.8%	85.1%	85.0%	84.6%
Electronic → Kitchen	85.0%	85.1%	86.8%	87.1%

Thus, they can be used to map between two domains. One example is the work by Blitzer *et al.* [11], who propose the structural correspondence learning (SCL) algorithm to exploit domain adaptation techniques for sentiment classification. As we mentioned above, the SCL algorithm first identifies the common features that are shared among different domains as the pivot features. It then uses unlabeled data and the pivot features from both source and target domains to find a mapping between the features from these domains, by which a common feature space is constructed. Extending this idea, Pan *et al.* [44] develop a spectral feature alignment (SFA) algorithm to align domain-specific words from different domains into unified clusters, with the help of domain-independent words as a bridge. Compared to SCL, SFA can discover a robust representation for cross-domain data by fully exploiting the relationship between the domain-specific and domain-independent words via simultaneously co-clustering them in a common latent space. Table 1.3 shows some experimental results on sentiment classification [44]. We can observe that the domain adaptation algorithms achieve better performance than the baseline without considering out-of-domain data. The bound indicates the performance of the gold standard, which is an in-domain classifier trained with labeled data from the target domain.

1.4 Multi-Task Learning Models

Transfer learning is focused on learning in the target domain, where the source domains are only used as auxiliary information. In contrast, multi-task learning seeks to improve the generalization performance of each learning task with the help of the other related tasks [14, 7, 53]. As we mentioned in the beginning of the chapter, most existing multi-task learning methods consider all tasks to have the same importance. In this setting, we review some existing methods for multi-task learning problem, which can be further categorized into five sets: common representation approach, task regularization approach, task clustering approach, hierarchical Bayesian approach and task relationship learning approach.

1.4.1 Common Representation Based Approach

The "representation" here mostly denotes data representation. Neural network is the earliest model in this category. Note that a multi-task neural network is just a conventional multilayer feed-forward neural network that capture the commonality of the tasks when learning. In a multi-task neural network, the hidden layer corresponds to common data representation after some linear or nonlinear transformation. Following this strategy, Liao and Carin [38] extend radial basis function networks for multi-task learning. In their work, the hidden layer is treated as a common representation for each task. Since the radial basis function network has an analytical solution, it can use the data points in multiple tasks to determine the form of the RBF function in the hidden layer, which can be learned via active learning.

Argyriou *et al.* [4, 5] propose a multi-task feature learning method to learn the common representation for multi-task learning under a regularization framework:

$$\xi(\mathbf{U}, \mathbf{A}) = \sum_{i=1}^{k} \sum_{j=1}^{n_i} l(y_j^i, \mathbf{a}_i^{\mathrm{T}} \mathbf{U}^{\mathrm{T}} \mathbf{x}_j^i) + \gamma ||\mathbf{A}||_{2,1}^2 \tag{1.3}$$

where $l(\cdot, \cdot)$ denotes the loss function, $\mathbf{U}$ is the common transformation to find common representation, $\mathbf{a}_i$ is the model parameters for task T_i, $\mathbf{A} = (\mathbf{a}_1, \ldots, \mathbf{a}_k)$ and $||\mathbf{A}||_{2,1} = \sum_j ||\mathbf{A}^j||_2$ denotes $l_{1,2}$ norm of a matrix $\mathbf{A}$, where $\mathbf{A}^j$ denotes jth row of $\mathbf{A}$. The $l_{1,2}$ norm in regularization function will lead to zero-row in $\mathbf{A}$, which is equivalent to feature selection on $\mathbf{U}^T \mathbf{x}_j^i$, because of the sparsity property of l_1 norm. An alternating method is used to learn the model parameters.

The common representation in multi-task neural network and multi-task feature learning is some form of transformation on the original data representation. Obozinski *et al.* [41] propose a feature selection method for multi-task learning, which formulation is similar to Eq. (1.3) but without $\mathbf{U}$. This can be viewed as a multi-task extension of LASSO [55]. Jebara [31] extends the maximum entropy discrimination (MED) method to multi-task learning [29]. MED solves the feature and kernel selection problems in multi-task learning settings, in which a selected subset of features and the kernel combination coefficients are shared by the tasks.

1.4.2 Task Regularization Approach

The task-regularization methods are all under the regularization framework, which consists of two objective function terms: an empirical loss on the training data of each task, and a regularization term that encodes the relationship between tasks for reducing the model complexity.

Evgeniou and Pontil [21] propose a multi-task extension of SVM, which

minimizes the following objective function

$$\xi(\{\mathbf{w}_i\}) = \sum_{i=1}^{k}\sum_{j=1}^{n_i} l(y_j^i, \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_j^i) + \lambda_1 \sum_{i=1}^{k} ||\mathbf{w}_i||_2^2 + \lambda_2 \sum_{i=1}^{k} ||\mathbf{w}_i - \frac{1}{k}\sum_{j=1}^{k}\mathbf{w}_j||_2^2. \quad (1.4)$$

The first and second terms of Eq. (1.4) denote the empirical error and 2-norm of parameter vectors, respectively, which are the same as those of single-task SVMs. However, the third term is designed to penalize large deviation between each parameter vector and the mean parameter vector of all tasks, which enforces the parameter vectors in all tasks are similar to each other.

In [20], Evgeniou *et al.* extend the work in [21] and propose multi-task kernel, by which the formulation of multi-task kernel methods can be reduced to that in single-task kernel methods.

Similar to [20], by utilizing an unweighted task network to encode the relatedness between tasks, Kato *et al.* [32] propose a different multi-task learning method, which is also based on SVM. The formulation can be written as

$$\begin{aligned} \min \quad & \xi(\{\mathbf{w}_i\}) = \sum_{i=1}^{k}\sum_{j=1}^{n_i} l(y_j^i, \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_j^i) + \lambda_1 \sum_{i=1}^{k} ||\mathbf{w}_i||_2^2 + \lambda_2\rho \\ s.t. \quad & ||\mathbf{w}_{i_k} - \mathbf{w}_{j_k}||_2^2 \leq \rho \text{ for } T_{i_k} \text{and } T_{j_k} \text{ are related,} \end{aligned}$$

which means the difference of the parameter vectors of any two related tasks is small.

1.4.3 Task Clustering Approach

Thrun and O'Sullivan [54] are the first to propose a task clustering method for multi-task learning. The main idea is to cluster all tasks into several clusters, in which the related tasks are assumed to share similar representations. The base learner in [54] is weighted K-Nearest-Neighbor classifier, in which each feature is given a weight for computing a distance metric, which are then used in clustering.

Different from [54], Bakker and Heskes [6] propose a Bayesian multi-task neural network, which has a structure that is the same to that of conventional multi-task neural network in which the input-to-hidden-layer weights are shared by all tasks. Different from the multi-task neural network, the hidden-layer-to-output-layer weights $\mathbf{A}_i$ for each task have a common prior.

Task-clustering methods require the number of clusters to be given, which is difficult for many real-world applications, Xue *et al.* [61] propose a task clustering multi-task learning method, which utilizes a nonparametric Bayesian model, Dirichlet Process (DP), as a basic mechanism to cluster tasks without knowing the number of clusters. For each task, they use logistic regression to model the data

$$p(y_j^i|\mathbf{x}_j^i, \mathbf{w}_i) = \sigma(y_j^i \mathbf{w}_i^{\mathrm{T}}\mathbf{x}_j^i).$$

Then we add a DP prior to $\mathbf{w}_i$ as

$$\mathbf{w}_i \sim \mathrm{DP}(\alpha_0, G_0)$$

where α_0 denotes the concentration parameter and G_0 denotes the base measure.

Researchers have also considered a task clustering approach under the regularization framework. Jacob *et al.* [30] propose a regularization method by incorporating the cluster structure as a regularization term. In [30], researchers introduce a cluster indicator matrix and integrate the cluster structure and empirical loss in the same objective function. A limitation of this method is that the number of clusters in multiple tasks must be given as a prior.

1.4.4 Hierarchical Bayesian Approach

Hierarchical Bayesian model is well studied in statistics community and widely used in many applications. Heskes [27] proposes a Bayesian multi-task neural network method for multi-task learning, in which the hidden-to-output weights for each task have a prior whose parameters are shared by all tasks. This model is similar to that in [6].

Micchelli and Pontil [40] are the first to employ a Gaussian Process (GP)[50] in multi-task learning. Lawrence and Platt [35] generalize the informative vector machine (IVM)[36], which is a sparse extension of GP, to multi-task learning. In this method, the parameters in the kernel function are shared by all tasks. Thus, the formulation in multi-task IVM is identical to that of single task IVM with the covariance matrix being a block matrix. Each sub diagonal matrix of this block matrix correspond to the covariance matrix for each task.

Yu *et al.* [64] propose a hierarchical Bayesian model, which utilize a GP for each task for multi-task regression. The nonparametric GP prior for each task is identical and the mean and covariance matrices have a conjugate prior in this model. Then, an EM algorithm is used to learn the mean and covariance matrix. Since the learned kernel matrix has no parametric form, when making a prediction, approximate estimation of kernel function is needed. Since all tasks share the same GP prior, the hierarchical model is affected by outlier tasks, which motivates a robust extension [66, 72] by utilizing a t-Process (TP) model. Different from the above methods, which are mostly based on GP, Zhang *et al.* [68] describe a latent variable model for multi-task learning. For each task T_i, the classifier or regressor is parameterized by some parameters $\boldsymbol{\theta}_i$. Then, the parameters $\boldsymbol{\theta}_i$ in different tasks are assumed to satisfy a latent variable model

$$\begin{aligned}\boldsymbol{\theta}_i &= \boldsymbol{\Lambda}\mathbf{s}_i + \mathbf{e}_i \\ \mathbf{e}_i &\sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}).\end{aligned}$$

From this formula, we can see $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$ are shared by all tasks. By changing

the probabilistic form of $\mathbf{s}_i$, this model is flexible to describe many variants in multi-task learning, such as independent tasks, noisy tasks, clusters of tasks, tasks having sparse representations, duplicated tasks and evolving tasks.

1.4.5 Task Relationship Learning Approach

In multi-task learning, a central issue is how to characterize the task relationships between different tasks. Most existing methods solve this problem by making an assumption on the task relationship; e.g., all task are similar or share the same data representation. Some methods utilize some *a priori* knowledge in some specific domains. However, in most cases, model assumption is hard to verify directly from the data. Moreover, in most applications, the *a priori* knowledge about task relationships does not exist. In these cases, we hope to learn task relationships from the data directly. Task clustering approaches can be viewed as a way to learn task relationships, although the learned relationship is only 'local' since they mostly ignore the negative correlations that may exist between different tasks in different task clusters. The multi-task GP model proposed in [12] is the first to learn the global task relationships in the form of a task covariance matrix. In the following, we briefly review this method.

The multi-task GP model in [12] directly models the task covariance matrix $\mathbf{\Sigma}$ by incorporating it into the GP prior, as follows:

$$\langle f_j^i, f_s^r \rangle = \Sigma_{ir} k(\mathbf{x}_j^i, \mathbf{x}_s^r), \tag{1.5}$$

where $\langle \cdot, \cdot \rangle$ denotes the covariance of two random variables, f_j^i is the latent function value for the jth data point $\mathbf{x}_j^i$ in the ith task, Σ_{ir} is the (i,r)th element of $\mathbf{\Sigma}$, and $k(\cdot, \cdot)$ is a kernel function. The output y_j^i given f_j^i is distributed as

$$y_j^i | f_j^i \sim \mathcal{N}(f_j^i, \sigma_i^2),$$

which defines the likelihood for $\mathbf{x}_j^i$. Here y_j^i is the label for $\mathbf{x}_j^i$ and σ_i^2 is the noise level of the ith task. One advantage of the formulation in [12] is its analytical form for the marginal likelihood. This is similar to conventional GP models where inference can be done efficiently. However, the model suffers from several drawbacks. One drawback is that when the number of tasks is large, the low-rank approximation used to reduce its computational cost may limit its expressive power. Another limitation is that, since the log-likelihood is non-convex with respect to $\mathbf{\Sigma}$ or to its low-rank approximation, the solution found by parameter-learning algorithms may be very sensitive to the initial value of $\mathbf{\Sigma}$ with no guarantee of the optimal solution.

To overcome the drawbacks of multi-task GP and also develop methods to learn the task relationships in other models, Zhang and Yeung develop a method called multi-task relationship learning method [71] that learns the

task relationship under the regularization framework:

$$\begin{aligned}\min_{\mathbf{W},\mathbf{b},\mathbf{\Omega}} \quad & \sum_{i=1}^{m}\frac{1}{n_i}\sum_{j=1}^{n_i}(y_j^i-\mathbf{w}_i^{\mathrm{T}}\mathbf{x}_j^i-b_i)^2+\frac{\lambda_1}{2}\mathrm{tr}(\mathbf{W}\mathbf{W}^{\mathrm{T}})+\frac{\lambda_2}{2}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^{\mathrm{T}}) \\ \text{s.t.} \quad & \mathbf{\Omega}\succeq 0 \\ & \mathrm{tr}(\mathbf{\Omega})\leq 1, \end{aligned} \tag{1.6}$$

where $\mathbf{w}_i$ and b_i are the model parameters for the ith task and $\mathbf{W} = (\mathbf{w}_1,\ldots,\mathbf{w}_m)$. This method can be viewed as maximum a posteriori (MAP) solution of the following probabilistic model:

$$\mathbf{W} \sim \left(\prod_{i=1}^{m}\mathcal{N}(\mathbf{w}_i \mid \mathbf{0}_d,\epsilon_i^2\mathbf{I}_d)\right) q(\mathbf{W}) \tag{1.7}$$

$$y_j^i \mid \mathbf{x}_j^i,\mathbf{w}_i,b_i \sim \mathcal{N}(\mathbf{w}_i^T\mathbf{x}_j^i+b_i,\varepsilon_i^2) \tag{1.8}$$

where $\mathcal{N}(\mathbf{m},\mathbf{\Sigma})$ denotes the multivariate (or univariate) normal distribution with mean $\mathbf{m}$ and covariance matrix (or variance) $\mathbf{\Sigma}$. The novelty lies in the prior $q(\mathbf{W})$ on $\mathbf{W}$ which belongs to matrix variate distribution [25]. When $q(\mathbf{W}) = \mathcal{MN}_{d\times m}(\mathbf{0},\mathbf{I}\otimes\mathbf{\Omega})$ where $\mathcal{MN}_{d\times m}(\mathbf{M},\mathbf{A}\otimes\mathbf{B})$ denotes the matrix variate normal distribution[2] with mean $\mathbf{M}\in\mathbb{R}^{d\times m}$, row covariance matrix $\mathbf{A}\in\mathbb{R}^{d\times d}$ and column covariance matrix $\mathbf{B}\in\mathbb{R}^{m\times m}$, the MAP solution will lead to problem (1.6). Here $\mathbf{\Omega}$ is the column covariance matrix of $\mathbf{W}$ where each column represents each task and hence $\mathbf{\Omega}$ can represent the task covariance. Moreover, when $q(\mathbf{W}) = \mathcal{MN}_{d\times m}(\mathbf{0},\mathbf{\Sigma}\otimes\mathbf{I})$, the MAP solution will become the multi-task feature learning formulation presented in [4, 5].

1.4.6 Application Examples of Multi-task Learning

Multi-task learning has many applications in machine learning areas, e.g., computer vision, information retrieval, Bioinformatics. We will review some of these applications in the following.

- *Face Recognition*: Heisele *et al.* [26] propose a multi-task learning method for face recognition. This method first detects the components of a face and then combines the component features and a whole face for face recognition. Lapedriza *et al.* [34] propose a multi-task feature extraction method for face recognition. In this method, face recognition is treated as a target task, while other face tasks such as facial expression recognition as complementary tasks to help improve the performance of face recognition. This method works by maximizing the mutual information between low-dimensional representation and subject labels in

[2]The probability density function is defined as $p(\mathbf{X}\,|\,\mathbf{M},\mathbf{A},\mathbf{B}) = \frac{\exp\left(-\frac{1}{2}\mathrm{tr}\left(\mathbf{A}^{-1}(\mathbf{X}-\mathbf{M})\mathbf{B}^{-1}(\mathbf{X}-\mathbf{M})^T\right)\right)}{(2\pi)^{md/2}|\mathbf{A}|^{m/2}|\mathbf{B}|^{d/2}}$.

face recognition while minimizing the mutual information between low-dimensional representation and labels in complementary tasks using a quadratic mutual information [56].

- *Image Classification*: Quattoni *et al.* [46] propose a method for image classification using a prototype representation. In this method, unlabeled data are used firstly to learn prototype representation, and then used to select prototypes learned in previous stage by learning from some previous supervised learning tasks. They then use the selected prototypes for the target task. Ahmed *et al.* [1] propose a method for visual recognition via using multi-task neural network, in which the target task and pseudo tasks share a common representation via a common hidden layer. This method also proposes to generate pseudo tasks for visual recognition tasks. Kienzle and Chellapilla [33] propose a biased regularization method for personalized handwriting recognition, in which the parameters of SVM in source tasks provide a bias of the target task. This bias is added to regularization term of SVM for target task as a prior.

- *Object Detection*: Torralba *et al.* [57] propose a method for multiclass object detection. Different from previous methods in object detection that train a classifier for individual object detection, this method solves multiclass object detection simultaneously by using shared features in multi-class objects, which can also reduce the number of features used in object detection.

- *Image Segmentation*: An *et al.* [2] utilize Dirichlet process and kernel stick-breaking process to segment multiple images simultaneously. Dirichlet process is used as a prior of base measure in kernel stick-breaking process and kernel stick-breaking process is used to incorporate the spatial information contained in images to help the segmentation and cluster image features in multiple images into several clusters to complete image segmentation. This work can be viewed as a way for multi-task clustering.

- *Collaborative Filtering*: Yu *et al.* [65] unify content-based filtering and collaborative filtering (CF) in a framework by using task clustering method, in which the parameters for each user profile share the same DP prior. Yu and Tresp [63] propose to use a multi-task learning method to solve the CF problem. In this model, the low-rank matrix approximation, which is used widely in CF, can be reformulated as a similar formulation in regularization framework in multi-task learning. The methods in [13, 70] utilize the useful information in multiple domains to improve the performance on each domain by learning domain relations in the form of a covariance matrix.

- *Age Estimation*: In [73], Yu and Yeung formulate the age estimation problem as a multi-task learning problem, where each task corresponds

to estimating ages based on the images of one person, and propose a multi-task extension of warped GP to solve this problem.

- *Text Classification*: Raina *et al.* [49] propose a transfer learning method for binary text classification problem. This method places a Gaussian prior on the parameters of logistic regression for target task and it learns the covariance matrix of the covariance matrix from source tasks. Do and Ng [19] also propose a logistic regression based method for text classification for multi-class problem.

- *Bioinformatics*: Xu *et al.* [60] use multi-task learning to solve the protein subcellular location prediction problem. Liu *et al.* [39] use multi-task feature learning method [4, 5] for cross-platform siRNA efficacy prediction, and Zhang *et al.* [69] identifies common mechanisms of responses to therapeutic targets. Puniyani *et al.* [45] utilize multi-task feature selection method on multi-population GWA mapping problems, and Lee *et al.* [37] extend multi-task feature selection method by learning the hyperparameters for solving the eQTL detection problem. Bickel *et al.* [8] provide a multi-task learning method based distribution matching for HIV therapy screening.

- *Finance*: Ghosn and Bengio [23] apply multi-task learning method for stock selection. Different from previous methods, which use one neural network to predict the return of one stock, the method in [23] learns several stocks in one neural network, in which the hidden layer is shared by all stocks and can be viewed as a common representation for all stocks. Experimental results show the generalization ability of multi-task neural network is much better than various benchmarks.

- *Robot Inverse Dynamics*: The methods in [15, 62] apply multi-task GP regression model in [12] for robot inverse dynamics that can improve the performance over previous methods.

1.5 Conclusion and Future Work

In this chapter, we have discussed transfer learning and multi-task learning frameworks and related them to cost sensitive learning. Transfer and multi-task learning approaches are useful when a learning problem is difficult to solve in a domains, but some related knowledge can be found in some other domains. In such cases, we may find some common knowledge between these domains to help improve the learning performance. We have systematically reviewed typical approaches to transfer and multi-task learning problems in inductive learning settings. In particular, we have pointed out that transfer

learning can be seen as a type of cost-sensitive learning where the costs are associated with the instances in both the source and target domains, and heavier weights can also be associated with the target domain.

When covariate shift assumption holds, many successful algorithms have been proposed. However, most of problems have not been solved when covariate shift assumption fails. First, it is unclear if there exist any weaker assumptions under which successfully transfer learning algorithms could be obtained. A future direction in this problem is how to combine labeled and unlabeled data to improve the estimation of the importance weight when covariate shift assumption does not hold but there is some labeled data available.

Besides, there could be more to explore on the multi-task learning from cost-sensitive learning perspective. In typical multi-task learning approaches so far, the task-associated costs such as the misclassification costs are placed equally on all tasks, and cost-sensitive learning is done in isolating the common knowledge between the tasks via many of the approaches that we reviewed. However, in some approach for multi-task learning, i.e., task relationship learning approach, the variance of each task, which is record in task covariance matrix, can be viewed as a cost for each task. Furthermore, it is also an open problem on how to place the costs on different tasks. The costs should reflect the relation between tasks. However, It is also unclear how to explore the structure in the relation between tasks to estimate the costs. Last but not least, how can cost sensitive learning bridge multi-task learning and transfer learning is also an interesting topic to explore more.

Bibliography

[1] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. P. Xing. Training hierarchical feed-forward visual recognition models using transfer learning from pseudo-tasks. In *Proceedings of the 10th European Conference on Computer Vision*, pages 69–82, Marseille, France, 2008.

[2] Q. An, C. Wang, I. Shterev, E. Wang, L. Carin, and D. B. Dunson. Hierarchical kernel stick-breaking process for multi-task image analysis. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning*, pages 17–24, Helsinki, Finland, 2008.

[3] R. K. Ando and T. Zhang. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine learning Research*, 6, 2005.

[4] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 41–48, Vancouver, British Columbia, Canada, 2006.

[5] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[6] B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4:83–99, 2003.

[7] J. Baxter. A Bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28(1):7–39, 1997.

[8] S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer. Multi-task learning for HIV therapy screening. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning*, pages 56–63, Helsinki, Finland, 2008.

[9] S. Bickel and T. Scheffer. Discriminative Learning Under Covariate Shift. *Journal of Machine Learning Research*, 10:2137–2155, 2009.

[10] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman. Learning bounds for domain adaptation. *Advances in Neural Information Processing Systems*, 20:129–136, 2007.

[11] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, Sydney, Australia, July 2006. Association for Computational Linguistics.

[12] E. Bonilla, K. M. A. Chai, and C. Williams. Multi-task Gaussian process prediction. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 153–160, Vancouver, British Columbia, Canada, 2007.

[13] B. Cao, N. N. Liu, and Q. Yang. Transfer learning for collective link prediction in multiple heterogenous domains. In *Proceedings of the 27th International Conference on Machine Learning*, pages 159–166, Haifa, Israel, 2010.

[14] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[15] K. M. A. Chai, C. K. I. Williams, S. Klanke, and S. Vijayakumar. Multi-task Gaussian process learning of robot inverse dynamics. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 265–272, Vancouver, British Columbia, Canada, 2008.

[16] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh. Sample selection bias correction theory. In *Proceedings of the International Conference on Algorithmic Learning Theory*. Springer, 2008.

[17] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Transferring naive bayes classifiers for text classification. In *Proceedings of The National Conference on Artifical Intelligence*, 2007.

[18] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, pages 193–200, Corvalis, Oregon, 2007. ACM Press.

[19] C. Do and A. Ng. Transfer learning for text classification. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 299–306, Vancouver, British Columbia, Canada, 2006.

[20] T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

[21] T. Evgeniou and M. Pontil. Regularized multi-task learning. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 109–117, Seattle, Washington, USA, 2004.

[22] Y. Freund. Boosting a weak learning algorithm by majority. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 202–216, 1990.

[23] J. Ghosn and Y. Bengio. Multi-task learning for stock selection. In M. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 946–952, Denver, CO, USA, 1996.

[24] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. *Journal of Machine Learning Research*, 1:1–10, 2008.

[25] A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*. Chapman & Hall, 2000.

[26] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 1239–1245, Vancouver, British Columbia, Canada, 2001.

[27] T. Heskes. Solving a huge number of similar tasks: A combination of multi-task learning and a hierarchical bayesian approach. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 233–241, Madison, Wisconson, USA, 1998.

[28] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf. Correcting Sample Selection Bias by Unlabeled Data. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 601–608. MIT Press, Cambridge, MA, 2007.

[29] T. Jaakkola, M. Meila, and T. Jebara. Maximum entropy discrimination. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 470–476, Denver, Colorado, USA, 1999.

[30] L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: A convex formulation. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 745–752, Vancouver, British Columbia, Canada, 2008.

[31] T. Jebara. Multi-task feature and kernel selection for SVMs. In *Proceedings of the Twenty-first International Conference on Machine Learning*, Banff, Alberta, Canada, 2004.

[32] T. Kato, H. Kashima, M. Sugiyama, and K. Asai. Multi-task learning via conic programming. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 737–744, Vancouver, British Columbia, Canada, 2007.

[33] W. Kienzle and K. Chellapilla. Personalized handwriting recognition via biased regularization. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pages 457–464, Pittsburgh, Pennsylvania, USA, 2006.

[34] À. Lapedriza, D. Masip, and J. Vitrì. On the use of independent tasks for face recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 2008.

[35] N. D. Lawrence and J. C. Platt. Learning to learn with the informative vector machine. In *Proceedings of the Twenty-first International Conference on Machine Learning*, Banff, Alberta, Canada, 2004.

[36] N. D. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: The informative vector machine. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 609–616, Vancouver, British Columbia, Canada, 2002.

[37] S. Lee, J. Zhu, and E. Xing. Adaptive multi-task lasso: with application to eQTL detection. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1306–1314. 2010.

[38] X. Liao and L. Carin. Radial basis function network for multi-task learning. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 795–802, Vancouver, British Columbia, Canada, 2005.

[39] Q. Liu, Q. Xu, V. W. Zheng, H. Xue, Z. Cao, and Q. Yang. Multi-task learning for cross-platform siRNA efficacy prediction: An in-silico study. *BMC Bioinformatics*, 11, 2010.

[40] T. P. Minka and R. W. Picard. Learning how to learn is learning with point sets. MIT Media Lab note, revised in 1999, 1997.

[41] G. Obozinski, B. Taskar, and M. Jordan. Multi-task feature selection. Technical report, Department of Statistics, University of California, Berkeley, June 2006.

[42] S. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[43] S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pages 677–682, 2008.

[44] S. J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen. Cross-domain sentiment classification via spectral feature alignment. *Proceedings of the 19th International Conference on World Wide Web*, 2010.

[45] K. Puniyani, S. Kim, and E. P. Xing. Multi-population GWA mapping via multi-task regularized regression. *Bioinformatics [ISMB]*, 26(12):208–216, 2010.

[46] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 2008.

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

[48] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pages 759–766, New York, NY, USA, 2007.

[49] R. Raina, A. Y. Ng, and D. Koller. Constructing informative priors using transfer learning. In *Proceedings of the Twenty-Third International Conference on Machine Learning*, pages 713–720, Pittsburgh, Pennsylvania, USA, 2006.

[50] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, Massachusetts, 2006.

[51] S. Si, D. Tao, and B. Geng. Bregman Divergence-Based Regularization for Transfer Subspace Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(7):929–942, July 2010.

[52] M. Sugiyama. Covariate Shift Adaptation by Importance Weighted Cross Validation. *Journal of Machine Learning Research*, 8:985–1005, 2007.

[53] S. Thrun. Is learning the n-th thing any easier than learning the first? In D. S. Touretzky, M. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 640–646, Denver, CO, 1995.

[54] S. Thrun and J. O'Sullivan. Discovering structure in multiple learning tasks: The TC algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning*, pages 489–497, Bari, Italy, 1996.

[55] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B(Methodological)*, 58(1):267–288, 1996.

[56] K. Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Reasearch*, 3:1415–1438, 2003.

[57] A. B. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 762–769, Washington, DC, USA, 2004.

[58] Y. Tsuboi, H. Kashima, S. Hido, S. Bickel, and M. Sugiyama. Direct density ratio estimation for large-scale covariate shift adaptation. In *Proceedings of the SIAM International Conference on Data Mining*, pages 443–454, Atlanta, Georgia, USA, 2008.

[59] V. N. Vapnik. *Statistical Learning Theory*. Wiley New York, 1998.

[60] Q. Xu, S. Pan, H. Xue, and Q. Yang. Multitask learning for protein subcellular location prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2010.

[61] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *Journal of Machine Learning Research*, 8:35–63, 2007.

[62] D.-Y. Yeung and Y. Zhang. Learning inverse dynamics by Gaussian process regression under the multi-task learning framework. In G. S. Sukhatme, editor, *The Path to Autonomous Robots*, pages 131–142. Springer, 2009.

[63] K. Yu and V. Tresp. Learning to learn and collaborative filtering. In *NIPS Workshop on Inductive Transfer: 10 Years Later*, Vancouver, British Columbia, Canada, 2005.

[64] K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian processes from multiple tasks. In *Proceedings of the Twenty-Second International Conference on Machine Learning*, pages 1012–1019, Bonn, Germany, 2005.

[65] K. Yu, V. Tresp, and S. Yu. A nonparametric hierarchical bayesian framework for information filtering. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 353–360, Sheffield, UK, 2004.

[66] S. Yu, V. Tresp, and K. Yu. Robust multi-task learning with t-processes. In *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, pages 1103–1110, Corvalis, Oregon, USA, 2007.

[67] B. Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the Twenty-First International Conference on Machine Learning*, Banff, Alberta, Canada, 2004.

[68] J. Zhang, Z. Ghahramani, and Y. Yang. Learning multiple related tasks using latent independent component analysis. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1585–1592, Vancouver, British Columbia, Canada, 2005.

[69] K. Zhang, J. W. Gray, and B. Parvin. Sparse multitask regression for identifying common mechanism of response to therapeutic targets. *Bioinformatics [ISMB]*, 26(12):97–105, 2010.

[70] Y. Zhang, B. Cao, and D.-Y. Yeung. Multi-domain collaborative filtering. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 725–732, Catalina Island, California, 2010.

[71] Y. Zhang and D.-Y. Yeung. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 733–742, Catalina Island, California, 2010.

[72] Y. Zhang and D.-Y. Yeung. Multi-task learning using generalized t process. In *Proceedings of the 13rd International Conference on Artificial Intelligence and Statistics*, pages 964–971, Chia Laguna Resort, Sardinia, Italy, 2010.

[73] Y. Zhang and D.-Y. Yeung. Multi-task warped gaussian process for personalized age estimation. In *Proceedings of the 23rd IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 2010.