Towards Enabling Learnware to Handle Unseen Jobs[*]

Yu-Jie Zhang, Yu-Hu Yan, Peng Zhao, Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210023, China
{zhangyj, yanyh, zhaop, zhouzh}@lamda.nju.edu.cn

Abstract

The learnware paradigm attempts to change the current style of machine learning deployment, i.e., user builds her own machine learning application almost from scratch, to a style where the previous efforts of other users can be reused, given a publicly available pool of machine learning models constructed by previous users for various tasks. Each learnware is a high-quality pre-trained model associated with its specification. Although there are many models in the learnware market, only a few, even none, may be potentially helpful for the current job. Therefore, how to identify and deploy useful models becomes one of the main concerns, which particularly matters when the user's job involves certain unseen parts not covered by the current learnware market. It becomes more challenging because, due to the privacy consideration, the raw data used for training models in the learnware market are inaccessible. In this paper, we develop a novel scheme that works can effectively reuse the learnwares even when the user's job involves unseen parts. Despite the raw training data are inaccessible, our approach can provably identify samples from the unseen parts while assigning the rest to proper models in the market for predicting under a certain condition. Empirical studies also validate the efficacy of our approach.

1 Introduction

Nowadays, machine learning models have shown impressive power in handling various jobs. However, when training a well-performed model, numerous data, fast machines, and expertise are always required, making it quite struggling to learn from scratch. The expensive cost has been a burden for users who hope to deploy learning models in their jobs. As there have already been vast well-performed models developed by different individuals, a natural question arises: Is it possible to build a model sharing platform, following a protocol to help the user benefit from existing models?

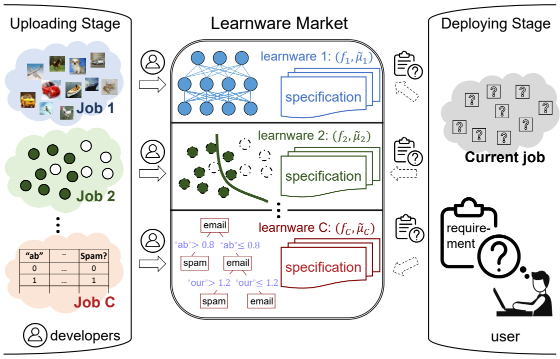


Figure 1: The learnware framework

Zhou (2016) considers running a learnware market shown as Figure 1. A learnware is a high-quality model associated with its specification served as an explanation or specialty for the model. There are two parts of participants in the market. The first is the developers, who could individually access the raw data for various learning jobs. Based on the local data, a develop can build a learnware and upload it to the market for sharing. The second part of participants is the users who hope to reuse the uploaded learnwares. When dealing with her own job, a user can search over the market, identify useful learnwares whose specifications matches the requirements, and then use her data to adapt/polish these learnwares for her job. The main focus here is how to design an effective protocol between developers and users, guiding the development and deployment of learnwares. The desired learnware should satisfy several properties (Zhou 2016).

Reusability is one of the core requirements concerning *which* and *how* the pre-trained models in the market can be reused. Although there are various models, the potentially useful ones could be few, even non-existent. Thus, it is important to identify *which* models are helpful for the user's current job. Meanwhile, as we cannot expect that there exists a model trained exactly for the current job, the identified models could only be partially helpful. Even worse, the current job might contain an unseen part, which is intractable to all existing models. Thus, *how* to fully exploit reusable models while leaving the intractable parts for post-processing is also a crucial problem. In addition, we hope the raw data are inaccessible to the user, which ensures developers can share their experience safely and avoid data privacy violations.

In this paper, we consider the reusability of learnware in the scenario where the user hopes to exploit the market to predict her current job directly. This problem is of interest

[*]This research was supported by NSFC (61921006)

when the user's training data are too scarce to support model training. Meanwhile, suppose the user has a label budget. In that case, the scheme can also be served as a preprocessing for identifying tractable samples while leaving the rest for label annotation, where the labeling efficiency can be improved.

It is challenging to ensure reusability and inaccessibility of raw data simultaneously, as the pre-trained model itself might not be informative enough to support reusability while raw data are inaccessible. The pioneering work (Wu et al. 2020) attempts to reuse models via their proposed reduced kernel mean embedding (RKME) specification. This method works effectively when the user's job covered well by the learnware market, whereas it remains a challenge to handle scenarios where the user's job involves unseen parts, which often occurs particularly when the learnware market is still in developing. In this paper, we designs a novel deploying approach to reuse models with RKME specification. Our method can provably identify samples from unseen parts of the current job and assign the rest to proper usable models, which provides answers to the "which" and "how" problems.

To facilitate an effective approach, we integrate several techniques from weakly supervised learning (Zhou 2018) with the RKME specification. Specifically, by tailoring the mixture proportion estimation technique (Ramaswamy, Scott, and Tewari 2016), we answer the problem of "which" by estimating the proportion of each uploaded job and the unseen part in the current job. Then, with the ratios, the risk rewriting technique (du Plessis, Niu, and Sugiyama 2014; Zhang et al. 2020) is used to train a job selector, which can directly identify samples from the unseen part and assign the rest to proper models. This addresses the "how" problem. Our method is theoretically supported by the convergence analysis of the mixture proportion estimator and generalization error analysis for the job selector. We also validate the effectiveness of our proposal by extensive experiments.

2 Preliminary

This section reviews the RKME specification, and related techniques, including KME and reduced set construction.

Kernel Mean Embedding. KME (Smola et al. 2007) makes a powerful representation for a probability distribution. The idea is to map a probability distribution to a reproducing kernel Hilbert space (RKHS), by representing each distribution P defined over $\mathcal{X}$ as a mean function

$$\mu_P := \int_{\mathcal{X}} k(\mathbf{x}, \cdot)\, \mathrm{d}P(\mathbf{x}),$$

where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a symmetric and positive definite kernel function (Schölkopf and Smola 2002), with associated RKHS $\mathcal{H}$ and feature map $\phi : \mathcal{X} \to \mathcal{H}$. The embedding μ_P exists and belongs to $\mathcal{H}$, when $\mathbb{E}_{\mathbf{x}\sim P}[\sqrt{k(\mathbf{x},\mathbf{x})}] < \infty$.

KME enjoys several favorable properties and is potentially a nice choice for the specification. First of all, KME makes a good representation of the distribution, as it can capture all information about the distribution when using the characteristic kernels such as the Gaussian kernel (Sriperumbudur, Fukumizu, and Lanckriet 2011). Meanwhile, several elementary operations on distribution can be easily performed with KME. For example, we can calculate the mean of any function $f \in \mathcal{H}$ over distribution P directly by the reproducing property, i.e., $\mathbb{E}_P[f(\mathbf{x})] = \langle f, \mu_P \rangle$ for all $f \in \mathcal{H}$. These excellent properties encourage KME's applications in various distribution related tasks (Gretton et al. 2012; Muandet and Schölkopf 2013; Doran 2013). For more about KME, we refer readers to the fantastic survey (Muandet et al. 2017).

In practice, we can only observe a dataset $D = \{\mathbf{x}_n\}_{n=1}^N$ sampled i.i.d. from the underlying distribution P. Thus, the empirical KME $\widehat{\mu}_P$ is calculated on the dataset to approximate the expected version μ_P as

$$\widehat{\mu}_P = \frac{1}{N}\sum\nolimits_{n=1}^{N} k(\mathbf{x}_n, \cdot).$$

As shown by Smola et al. (2007), the empirical KME $\widehat{\mu}_P$ converges to μ_P in the rate of $O(1/\sqrt{N})$ measured by the RKHS norm $\|\cdot\|_{\mathcal{H}}$ under mild conditions.

Reduced Kernel Mean Embedding. Although enjoying several nice properties, KME requires accessing to raw data, which violates the inaccessibility of learnware and is not a valid specification. To address this issue, Wu et al. (2020) introduce the reduced KME to approximated the original KME via the reduced set method. This method is first used to speed up SVM prediction (Burges 1996) and receives more comprehensive studies in Schölkopf et al. (1999).

The idea of RKME is to find a reduced set $\{(\beta_m, \mathbf{z}_m)\}_{m=1}^M$ whose KME $\widetilde{\mu}_P = \sum_{m=1}^M \beta_m k(\mathbf{z}_m, \cdot)$ approximates that of the original data $\{\mathbf{x}_n\}_{n=1}^N$. This is achieved by solving

$$\min_{\boldsymbol{\beta},\mathbf{z}} \left\| \frac{1}{N}\sum\nolimits_{n=1}^{N} k(\mathbf{x}_n, \cdot) - \sum\nolimits_{m=1}^{M} \beta_m k(\mathbf{z}_m, \cdot) \right\|_{\mathcal{H}}^2, \quad (1)$$

where $\beta_m \in \mathbb{R}$ is the coefficient and $\mathbf{z}_m \in \mathcal{X}$ is the sample of the reduced set. The above problem is known as the reduced set construction (Schölkopf et al. 1999), when $\mathbf{z}_i$ is a newly constructed sample. Several methods can be used for handling the above problem. We defer descriptions in Appendix A.

RKME $\widetilde{\mu}_P$ enjoys a linear convergence rate $O(e^{-M})$ to empirical KME $\widehat{\mu}_P$ when $\mathcal{H}$ is finite-dimensional (Bach, Lacoste-Julien, and Obozinski 2012), which makes it a good approximation of the distribution. Meanwhile, thanks to the reduced set construction, raw data are inaccessible to users. Thus, RKME is an effective specification.

3 Problem Setup

This section introduces the problem setup and notations. We denote joint, conditional, and marginal distributions by subscripts XY, $X|Y$, and X throughout the paper. The learnware protocol contains uploading and deploying stages.

Uploading Stage. In this stage, developers should upload well-performed models with RKMEs for their jobs.

Specifically, suppose there are C developers in the uploading stage. The i-th developer can access to a local dataset $D_i = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N_i}$ sampled from $\mathcal{D}_{XY}^i$, which is the joint distribution of the i-th job defined over $\mathcal{X}_i \times \mathcal{Y}_i$.

Based on dataset D_i, the developer can train a high-quality model $f_i : \mathcal{X}_i \to \mathcal{Y}_i$, which performs well over $\mathcal{D}_{XY}^i$. Besides, a RKME $\widetilde{\mu}_i = \sum_{m=1}^{M_i} \beta_m \cdot k(\mathbf{z}_m, \cdot)$ is constructed to approximate the empirical KME $\widehat{\mu}_i = 1/N_i \cdot \sum_{n=1}^{N_i} k(\mathbf{x}_n, \cdot)$ by minimizing (1). The RKME specification $\widetilde{\mu}_i$ makes a good approximation for the feature distribution $\mathcal{D}_X^i$ without accessing the raw data. Afterward, she can upload the learnware pair $(f_i, \widetilde{\mu}_i)$ to the market. For simplicity, we assume the same feature space for all jobs, i.e., $\mathcal{X}_i = \mathcal{X}$ for all $i \in [C]$.

Deploying Stage. In this stage, the user hopes to exploit learnwares $\{(f_i, \widetilde{\mu}_i)\}_{i=1}^{C}$ in the market to handle her own job. Specifically, we consider the scenario where the user hopes to obtain a classifier f to predict her dataset $D_{te} = \{\mathbf{x}_n\}_{n=1}^{N_t}$, whose labels are unknown but sampled from the distribution $\mathcal{D}_{XY}^{te}$ defined over $\mathcal{X} \times \mathcal{Y}$. It is almost impossible to handle this problem without any assumption since the distribution $\mathcal{D}_{XY}^{te}$ can be arbitrarily different from those of uploaded jobs. Recently, Wu et al. (2020) introduce the *instance-recurrent* assumption which assumes the user's job is well covered by the learnware market and $\mathcal{D}_X^{te}$ to be a mixture of $\mathcal{D}_X^i$, i.e.,

$$\mathcal{D}_{XY}^{te} = \sum\nolimits_{i=1}^{C} w_i \cdot \mathcal{D}_{XY}^{i}, \tag{2}$$

where $\mathbf{w} \in \Delta_C$. It remains challenging to consider the existence of unseen parts in the user's job, which often happens particularly when the market is still in developing.

In this paper, we generalize the previous assumption (2) by further considering the unseen parts in the user's job, which are not covered by the market and have the distribution $\mathcal{D}_{XY}^{\mathsf{u}}$ over $\mathcal{X} \times \mathcal{Y}_{\mathsf{u}}$. Since the user's data provide no supervision, $\mathcal{Y}_{\mathsf{u}}$ is assumed to have a unique class as $\mathcal{Y}_{\mathsf{u}} = \{\mathsf{u}\}$. We refer unseen parts of the user's job as an *unseen job* and introduce the following unseen-job assumption.

Assumption 1 (unseen-job assumption)**.** The distribution of the user's job $\mathcal{D}_{XY}^{te}$ is a mixture of those of uploaded jobs $\mathcal{D}_{XY}^{i}$ and the unseen job $\mathcal{D}_{XY}^{\mathsf{u}}$ as

$$\mathcal{D}_{XY}^{te} = \sum\nolimits_{i=1}^{C} w_i \mathcal{D}_{XY}^{i} + w_{\mathsf{u}} \mathcal{D}_{XY}^{\mathsf{u}}, \tag{3}$$

where $\sum_{i=1}^{C} w_i + w_{\mathsf{u}} = 1$, $w_i, w_{\mathsf{u}} \geq 0$ for all $i \in [C]$.

Assumption 1 is a natural generalization of the instance-recurrent assumption of Wu et al. (2020), since (3) recovers (2) when $w_{\mathsf{u}} = 0$. Due to the existence of the unseen job, our problem becomes much harder to solve and requires a new deploying approach. We discuss potential directions that can further generalize Assumption 1 in Appendix A.

4 Our Approach

This section describes our approach. The idea is to train a selector to recognize which job a user's testing sample comes from. With a well-performed job selector, we can identify intractable samples belonging to the unseen job while assigning the rest to proper uploaded models for predicting, which can thereby address the problems of "which", and "how" proposed in Section 1. The challenge is how to train the selector when the unseen job could exist and raw data of uploaded jobs are inaccessible.

4.1 Desired Job Selector

To train the job selector, we take each job as a superclass, whose label is denoted by i for the i-th job and u for the unseen job. Thus, the selector is essentially a multiclass classifier $g(\mathbf{x}) : \mathcal{X} \to \mathcal{I}$ with $\mathcal{I} = \{1, \ldots, C, \mathsf{u}\}$. Given a testing sample $\mathbf{x}$, when $g(\mathbf{x}) \in [C]$, the user can predict it as $f(\mathbf{x}) = f_{g(\mathbf{x})}(\mathbf{x})$ with the $g(\mathbf{x})$-th uploaded model, otherwise predict that it comes from the unseen job as $f(\mathbf{x}) = \mathsf{u}$. We show that the generalization error of the decision function f is highly related to the quality of the job selector.

Proposition 1. *Under Assumption 1, suppose all uploaded models $\{f_i\}_{i=1}^{C}$ perform well over their jobs such that $\mathbb{E}_{\mathcal{D}_{XY}^i}[\mathbb{1}(f_i(\mathbf{x}) \neq y)] \leq \epsilon$ holds for all $i \in [C]$. The generalization error of the prediction function f is bounded by*

$$\mathbb{E}_{\mathcal{D}_{XY}^{te}}[L_{01}(f(\mathbf{x}), y)] \leq \epsilon + R(g), \tag{4}$$

where $L_{01}(f(\mathbf{x}), y) = \mathbb{1}[f(\mathbf{x}) \neq y]$ and $\mathbb{1}[\cdot]$ is the indicator function. The risk of job selector $R(g)$ is defined by

$$\sum\nolimits_{i=1}^{C} w_i \mathbb{E}_{\mathcal{D}_X^i}[L_{01}(g(\mathbf{x}), i)] + w_{\mathsf{u}} \mathbb{E}_{\mathcal{D}_X^{\mathsf{u}}}[L_{01}(g(\mathbf{x}), \mathsf{u})] \tag{5}$$

Proposition 1 implies that once the selector g minimizes $R(g)$, we can obtain a high-quality decision function by exploiting the uploaded models. We note that minimizing (5) is the key to address the "which" and "how" problems, as the mixture proportions $\{w_i\}_{i=1}^{C}$ indicate which models are useful and the selector tells how to use them. Unfortunately, the 0-1 loss L_{01} is non-convex, which makes the risk minimization problem generally intractable.

The common practice to handle this problem is to use a convex surrogate loss function to substitute the 0-1 loss. In the multiclass scenario, instead of training a single classifier g directly, we usually train multiple binary classifiers $g_i : \mathcal{X} \to \mathbb{R}$ for each $i \in \mathcal{I}$ to distinguish the i-th class from others, and then predict as $g(\mathbf{x}) = \arg\max_{i \in \mathcal{I}} g_i(\mathbf{x})$. Denoting by $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), \ldots, g_C(\mathbf{x}), g_{\mathsf{u}}(\mathbf{x})]$ the vector function, we can train the binary classifiers by minimizing the expected risk $R_\Psi(\mathbf{g})$ w.r.t. a convex surrogate loss defined by

$$\sum\nolimits_{i=1}^{C} w_i \mathbb{E}_{\mathcal{D}_X^i}[L_\Psi(\mathbf{g}(\mathbf{x}), i)] + w_{\mathsf{u}} \mathbb{E}_{\mathcal{D}_X^{\mathsf{u}}}[L_\Psi(\mathbf{g}(\mathbf{x}), \mathsf{u})] \tag{6}$$

There are multiple choices for the multiclass surrogate loss $L_\Psi : \mathbb{R}^{C+2} \to \mathbb{R}$ (Zhang 2004), such as the one-versus-rest (OVR) and pairwise comparison (PC) losses. The consistency between functions minimizing $R_\Psi(\mathbf{g})$ and that minimizing $R_\Psi(g)$ is also studied in the seminal work of Zhang (2004).

To minimize the expected risk $R_\Psi(\mathbf{g})$, it requires knowing $\mathcal{D}_X^i$, $\mathcal{D}_X^{\mathsf{u}}$, $\{w_i\}_{i=1}^{C}$ and w_{u}. The feature distribution $\mathcal{D}_X^i$ can be approximated directly by the RKMEs $\widetilde{\mu}_i$. However, the estimation of $\mathcal{D}_X^{\mathsf{u}}$ is challenging since the unseen job has never been uploaded in the learnware market. In the next part, we present the method to minimize R_Ψ with only the RKMEs $\{\widetilde{\mu}_i\}_{i=1}^{C}$ and user's testing data, assuming that the mixture proportions $\{w_i\}_{i=1}^{C}$, w_{u} were known, for a moment.

4.2 Expected Risk Rewriting

One of the main challenges to minimize $R_\Psi(\mathbf{g})$ is that the learnware market cannot provide any information about the marginal distribution of unseen job $\mathcal{D}_X^{\mathsf{u}}$. We handle this problem by using the risk rewriting technique (du Plessis, Niu, and Sugiyama 2014; Zhang et al. 2020) with the user's unlabeled data. The intuition is that though the unseen job is never uploaded, its marginal distribution $\mathcal{D}_X^{\mathsf{u}}$ is hidden in the user's testing data. Under Assumption 1, summing over the label set, we can estimate $\mathcal{D}_X^{\mathsf{u}}$ by separating the distributions of uploaded job from that of the unlabeled data as

$$\mathcal{D}_X^{\mathsf{u}} = \big(\mathcal{D}_X^{te} - \sum\nolimits_{i=1}^{C} w_i \mathcal{D}_X^i\big)/w_{\mathsf{u}}. \tag{7}$$

Rewriting (6) with (7), we have the following proposition.

Proposition 2. *Under Assumption 1, for all measurable function $g_i : \mathcal{X} \to \mathbb{R}$ with $i \in \mathcal{I}$, we have*

$$\begin{aligned} R_\Psi(\mathbf{g}) = \sum\nolimits_{i=1}^{C} w_i \mathbb{E}_{\mathcal{D}_X^i}[L_\Psi(\mathbf{g}(\mathbf{x}), i) - L_\Psi(\mathbf{g}(\mathbf{x}), \mathsf{u})] \\ + \mathbb{E}_{\mathcal{D}_X^{te}}[L_\Psi(\mathbf{g}(\mathbf{x}), \mathsf{u})]. \end{aligned} \tag{8}$$

Proposition 2 show that $R_\Psi(\mathbf{g})$ is only established on the distribution of uploaded jobs $\mathcal{D}_X^i$ and that of the user's data $\mathcal{D}_X^{te}$. Thus, we can train the selector by minimizing its empirical version $\widehat{R}_\Psi(\mathbf{g})$, where the distributions $\mathcal{D}_X^i$ and $\mathcal{D}_X^{te}$ are approximated with their empirical observations D_i and D_{te}.

However, raw data D_i are inaccessible and the only available resource is the associated RKME. To address this problem, we employ the *kernel herding technique* (Chen, Welling, and Smola 2010; Bach, Lacoste-Julien, and Obozinski 2012) to sample a mimic dataset $\widetilde{D}_i = \{\mathbf{x}_i\}_{n=1}^{\widetilde{N}_i}$ from the RKME $\widetilde{\mu}_i$ to substitute D_i, where the empirical risk $\widehat{R}_\Psi(\mathbf{g})$ becomes

$$\begin{aligned} \widehat{R}_\Psi(\mathbf{g}) = & \sum\nolimits_{i=1}^{C} \frac{w_i}{\widetilde{N}_i} \sum_{\mathbf{x}_n \in \widetilde{D}_i} (L_\Psi(\mathbf{g}(\mathbf{x}_n), i) - L_\Psi(\mathbf{g}(\mathbf{x}_n), \mathsf{u})) \\ & + \frac{1}{N_t} \sum_{\mathbf{x}_n \in D_{te}} L_\psi(\mathbf{g}(\mathbf{x}_n), \mathsf{u}). \end{aligned} \tag{9}$$

In Section 5, we theoretically justify that $\widetilde{D}_i$ provides a sufficient approximation for training the selector.

The last issue is that $\widehat{R}_\Psi(\mathbf{g})$ is generally non-convex, making it hard to optimize. We can eliminate the non-convexity by choosing proper multi-class surrogate losses L_Ψ and inner binary losses ψ. In this paper, We use the OVR loss

$$L_\Psi^{\mathtt{OVR}}(\mathbf{g}(\mathbf{x}), i) = \psi(g_i(\mathbf{x})) + \sum\nolimits_{j \neq i} \psi(-g_j(\mathbf{x})),$$

with the convex binary loss $\psi : \mathbb{R} \to \mathbb{R}$ satisfying $\psi(z) - \psi(-z) = -z$, for all $z \in \mathbb{R}$ (du Plessis, Niu, and Sugiyama 2015).[1] In such a case, $\widehat{R}_\Psi(\mathbf{g})$ is convex w.r.t. $\mathbf{g}$ and can be optimized efficiently. Practically, we train the selector by

$$\widetilde{\mathbf{g}} = \arg\min\nolimits_{g_1,\ldots,g_C,g_{\mathsf{u}} \in \mathcal{G}} \widehat{R}_\Psi(\mathbf{g}) \tag{10}$$

where $\mathcal{G} = \{g \in \mathcal{H} \mid \|g\|_{\mathcal{H}} \leq B_{\mathcal{G}}\}$ is a RKHS-based hypothesis set. We present more descriptions for kernel herding and the convex formulation of $\widehat{R}_\Psi(\mathbf{g})$ in Appendix A.

4.3 Mixture Proportion Estimation with RKME

This part shows how to estimate $\{w_i\}_{i=1}^C$ and $\{w_{\mathsf{u}}\}$ with RKMEs and user's unlabeled data.

No Unseen Job. The problem is straightforward suppose we knew there is no unseen job. By summing (2) over the label set $\mathcal{Y}$, we have $\mathcal{D}_X^{te} = \sum_{i=1}^C w_i \mathcal{D}_X^i$. Since the RKME $\widetilde{\mu}_i$ is a good approximation of the marginal distribution $\mathcal{D}_X^i$, we use the relationship among KMEs to solve $\{w_i\}_{i=1}^C$ by

$$\min_{\mathbf{w} \in \Delta} \left\| \sum\nolimits_{i=1}^{C} w_i \widetilde{\mu}_i - \widehat{\mu}_{te} \right\|_{\mathcal{H}}^2, \tag{11}$$

where $\mathbf{w} = [w_1, \cdots, w_C]^{\mathrm{T}}$ and Δ_C is the probability simplex. Notation $\widehat{\mu}_{te} = 1/N_t \cdot \sum_{n=1}^{N_t} k(\mathbf{x}_n, \cdot)$ is the empirical KME of the testing data. Note that (11) can be solved by a quadratic program (Smola et al. 2007).

The above method is *infeasible* in our setting due to the unseen job's absence. Our idea is to estimate individual mixture proportion w_i and then calculate $w_{\mathsf{u}} = 1 - \sum_{i=1}^C w_i$.

MPE Problem. The estimation of individual w_i can be cast as a mixture proportion estimation (MPE) problem (Blanchard, Lee, and Scott 2010; Jain et al. 2016; du Plessis, Niu, and Sugiyama 2017). In the MPE problem, the user observes two datasets, $D_F = \{\mathbf{x}_n\}_{n=1}^{N_f}$ sampled i.i.d. from distribution F and $D_H = \{\mathbf{x}_n\}_{n=1}^{N_h}$ sampled i.i.d from H, where $F = wH + (1-w)G$ is a mixture of H and an unobserved distribution G with proportion $w \in [0,1]$. The goal of the user is to estimate w with the empirical data D_F and D_H.

Clearly, in our scenario, we can take $F = \mathcal{D}_X^{te}$ and $H = \mathcal{D}_X^i$, and use D_{te} and D_i to estimate w_i with developed estimators. Nevertheless, the raw training data D_i are inaccessible, which limits the application of most existing MPE estimators as they always require the raw data. To handle this problem, we tailor the KME based estimator (Ramaswamy, Scott, and Tewari 2016) to fit the RKME specification.

Expected KME Estimator. MPE problem is generally ill-defined unless we impose certain assumptions. In literature, the irreducible assumption and its variants (Blanchard, Lee, and Scott 2010; Scott 2015) are proposed to ensure a unique value of the true mixture proportion w_i. Furthermore, Blanchard, Lee, and Scott (2010) have observed that the true w_i is identical to the maximum proportion of $\mathcal{D}_X^i$ in $\mathcal{D}_X^{te}$ when the irreducible assumption holds.

Based on the observation, we can identify w_i by the maximum $\widehat{w}_i$ that makes $G' = (\mathcal{D}_X^{te} - \widehat{w}_i \mathcal{D}_X^i)/(1 - \widehat{w}_i)$ still a valid distribution. That is, if the estimated mixture proportion $\widehat{w}_i$ is greater than the true one w_i, the density function of G' would go negative, making it an illegal distribution. The problem here is how to judge whether G' is a valid distribution?

The above problem can be handled with KME. For simplicity, we rewrite the mixture proportion as $\lambda_i = 1/(1-w_i)$ and $\widehat{\lambda}_i = 1/(1-\widehat{w}_i)$, where $\lambda_i \in [1, +\infty)$ is monotonically increasing w.r.t. w_i. In such a case, $G' = \widehat{\lambda}_i \mathcal{D}_X^{te} + (1-\widehat{\lambda}_i)\mathcal{D}_X^i$. One can define the distance from G' to a set containing all valid distributions as

$$d(\widehat{\lambda}_i) = \inf_{h \in \mathcal{C}} \big\| \underbrace{\widehat{\lambda}_i \mu_{te} + (1-\widehat{\lambda}_i)\mu_i}_{\mu_{G'}} - h \big\|_{\mathcal{H}},$$

[1] Many loss functions satisfy the condition, such as logistic loss $\psi(z) = \log(1 + \exp(-z))$, square loss $\psi(z) = (1-z)^2/4$ and double hinge loss $\psi(z) = \max(-z, \max(0, (1-z)/2))$

where $\mathcal{C} = \{h \in \mathcal{H} \mid h = \mu_Q$ for some distribution $Q\}$ is the set of KMEs of all valid distributions. If $\widehat{\lambda}_i \le \lambda_i$, G' is a valid distribution, then $d(\widehat{\lambda}_i) = 0$, otherwise $d(\widehat{\lambda}_i) > 0$. Moreover, the distance function $d(\widehat{\lambda}_i)$ is non-decreasing and convex over $[1, +\infty)$. Thus, we can use binary search to identify λ_i as the critical value $\widehat{\lambda}_i$ making $d(\widehat{\lambda}_i) > 0$. More illustration of the idea and formal descriptions of properties of $d(\widehat{\lambda})$ can be found in Appendix B.

Empirical RKME Estimators. In the deploying phase, we can approximate $d(\widehat{\lambda}_i)$ empirically by

$$\widehat{d}(\widehat{\lambda}_i) = \inf_{h \in \widehat{\mathcal{C}}} \left\| \widehat{\lambda}_i \widehat{\mu}_{te} + (1 - \widehat{\lambda}_i)\widetilde{\mu}_i - h \right\|_{\mathcal{H}}, \qquad (12)$$

where $\widehat{\mu}_{te}$ is the empirical KME of the testing data and $\widetilde{\mu}_i$ is the RKME for the i-th job. The set $\widehat{\mathcal{C}} = \{h \in \mathcal{H} \mid h = \sum_{n=1}^{N_t} a_n k(\mathbf{x}_n, \cdot) + b_m \sum_{m=1}^{M_i} k(\mathbf{z}_m, \cdot)$, for $\sum_{n=1}^{N_t} a_n + \sum_{m=1}^{M_i} b_m = 1\}$ contains all empirical KMEs established over $\{\mathbf{x}_n\}_{n=1}^{N_t}$ and $\{\mathbf{z}_m\}_{m=1}^{M_i}$. We can calculate the value of $(\widehat{d}(\widehat{\lambda}_i))^2$ by solving a quadratic program.

The empirical distance $\widehat{d}(\widehat{\lambda}_i)$ does not enjoy the nice properties as $d(\widehat{\lambda}_i)$, as it could be greater than 0, even when $\widehat{\lambda}_i < \lambda_i$. However, it is still non-decreasing and convex over $[1, +\infty)$ and converges to $d(\widehat{\lambda}_i)$ with the growth of N_t and M_i. We can specify a threshold ν rather than 0 to identify the true weight. Formally speaking, we can estimate λ_i by

$$\widehat{\lambda}_i^V = \inf\{\widehat{\lambda}_i : \widehat{d}(\widehat{\lambda}_i) > \nu\}. \qquad (13)$$

Besides, since $\widehat{d}(\widehat{\lambda}_i)$ is convex over $[1, +\infty)$, its gradient is non-decreasing over $[1, +\infty)$ and thus thresholding the gradient is also a viable strategy. We can estimate λ_i by

$$\widehat{\lambda}_i^G = \inf\{\widehat{\lambda}_i : \exists g' \in \partial \widehat{d}(\widehat{\lambda}_i), g' \ge \nu\}. \qquad (14)$$

Empirical studies show that $\widehat{\lambda}_i^G$ achieves better performance. Thus, we use it for our mixture proportion estimator. After estimating $\widehat{\lambda}_i^G$, we calculate $\widehat{w}_i^G = (\widehat{\lambda}_i^G - 1)/\widehat{\lambda}_i^G$.

4.4 A Summary of Deploying Approach

Finally, we summarize our deploying approach, where we first estimate the mixture proportion w_i by the estimator $\widehat{w}_i^G$ and calculate $\widehat{w}_{\text{u}}^G = 1 - \sum_{i=1}^{C} \widehat{w}_i^G$, followed by the selector training via (10). Then, we use the selector $\widetilde{g}(\mathbf{x}) = \arg\min_{i \in \mathcal{I}} \widetilde{g}_i(\mathbf{x})$ to identify samples from the unseen job and assign the rest to proper models for predicting. Detailed implementations of the MPE estimator (Algorithm 1) and selector training (Algorithm 2) are provided in Appendix A.

5 Theoretical Analysis

In this section, we theoretically justify our method by providing the excess risk bound for the selector and convergence analysis for the RKME based MPE estimator.

5.1 Excess Risk Bound for Job Selector

We first provide the excess risk bound for our selector trained with the true mixture proportions. The convergence analysis for the MPE estimator is shown in the next subsection. For simplicity, we further assume $M_i = M$ and $N_i = N$ for all $i \in [C]$ and the size of mimic data $|\widetilde{D}_i|$ is greater than M as we can generate an arbitrary number of mimic data.

Theorem 1. *Suppose $k(\mathbf{x}, \mathbf{x}) \le 1$ holds for all $\mathbf{x} \in \mathcal{X}$ and binary loss ψ satisfies $B_\psi \ge 0$ and is L-Lipschitz.*[2] *Then,*

$$R_\Psi(\widetilde{\mathbf{g}}) - R_\Psi(\mathbf{g}^*) \le O\big((C+1)(N^{-\frac{1}{2}} + N_t^{-\frac{1}{2}}) + M^{-\frac{1}{2}}\big)$$

holds for all $g_1, \ldots, g_C, g_u \in \mathcal{G}$. The optimal classifier $\mathbf{g}^* = \arg\min_{g_1, \ldots, g_C, g_u \in \mathcal{G}} R_\Psi(\mathbf{g})$ minimizes the expected risk R_Ψ over the hypothesis space $\mathcal{G}$.

Theorem 1 essentially states that despite the inaccessibility of the raw data and the existence of the unseen job, our job selector converges to the optimal one trained over the testing distribution. We defer the detailed proof to Appendix B.

Remark 1. The proof of Theorem 1 is established on the cornerstone that RKME $\widetilde{\mu}_i$ converges to the empircal KME $\widehat{\mu}_i$ in the rate of $O(1/\sqrt{M})$. We note that this rate can be improved to $O(e^{-M})$ when the RKHS is finite-dimensional (proofs are shown in Appendix B). This rate indicates that with a tiny reduced set $M = O(\log N)$, our selector $\widetilde{\mathbf{g}}$ converges to the optimal one $\mathbf{g}^*$ in the rate of $O((C+1)(1/\sqrt{N} + 1/\sqrt{N_t}))$, which is the standard rate for the empirical risk minimization method trained with raw data, which further validates the effectiveness of the RKME specification. Although in the infinite-dimensional setting, the current theory only supports the $O(1/\sqrt{M})$ rate, our experiments show that a tiny reduce set ($M = 10$ vs $N = 2500$) is sufficient to support satisfactory empirical performance.

5.2 Convergence Analysis for MPE Estimator

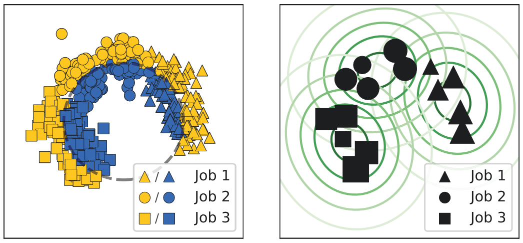
As we have mentioned, the MPE problem is not well defined unless we introduce certain assumptions. For analyzing the RKME based estimator, we require the separability assumption introduced by Ramaswamy, Scott, and Tewari (2016).

Assumption 2 (separability assumption)**.** A kernel k and distributions G, H satisfy the separability condition with margin $\alpha > 0$ and tolerance β, if $\exists h \in \mathcal{H}$, $\|h\|_{\mathcal{H}} < 1$ and

$$\mathbb{E}_{\mathbf{x} \sim G}[h(\mathbf{x})] \le \inf_{\mathbf{x}} h(\mathbf{x}) + \beta \le \mathbb{E}_{\mathbf{x} \sim H}[h(\mathbf{x})] - \alpha.$$

The separability assumption states that H and G are separated enough evaluated by a function space, which is a natural extension of the assumption supp$(H) \not\subseteq$ supp(G) to the function space setting (Ramaswamy, Scott, and Tewari 2016). In our case, $H = \mathcal{D}_X^i$ and $G = \bar{\mathcal{D}}_X^i := (\mathcal{D}_X^{te} - w_i \mathcal{D}_X^i)/(1 - w_i)$ is the rest distribution in $\mathcal{D}_X^{te}$ apart from $\mathcal{D}_X^i$. The two distributions are separable in applications as they are the marginal distribution for different jobs. We provide the convergence rate for estimator $\widehat{\lambda}_i^G$. The proof is deferred to Appendix B.

[2] Common surrogate loss functions satisfy these conditions, such as logistic loss, exp loss and square loss.



(a) Uploaded jobs (b) RKMEs

Figure 2: Uploading phase

Theorem 2. *Suppose $k(\mathbf{x}, \mathbf{x}) \leq 1$ holds for all $\mathbf{x} \in \mathcal{X}$. Let kernel k and distribution $\mathcal{D}_X^i$, $\bar{\mathcal{D}}_X^i$ satisfy the separability condition with margin $\alpha > 0$ and tolerance β. Let $\nu \in \left[\frac{\alpha}{4\lambda_i}, \frac{3\alpha}{4\lambda_i}\right]$, $\sqrt{\min\{N, M, N_t\}} \geq \frac{192\sqrt{\log(1/\delta)}}{\alpha/\lambda_i - \nu}$ and $\widetilde{\mu}_i \in \widehat{\mathcal{C}}$. With probability at least $1 - 4\delta$, we have*

$$\lambda_i - \widehat{\lambda}_i^G \leq O((\min\{N, M, N_t\})^{-1/2})$$
$$\widehat{\lambda}_i^G - \lambda_i \leq 8\beta\lambda_i/\alpha + O((\min\{N, M, N_t\})^{-1/2}).$$

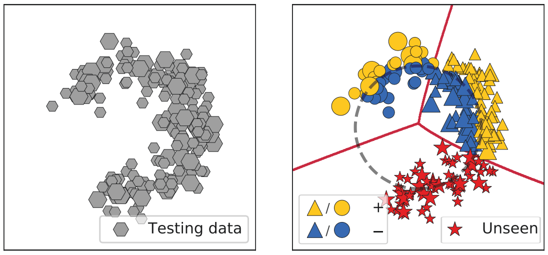
The convergence rate of MPE estimator is similar with the excess risk bound of job selector, whose dependence on M can also be improved to $O(e^{-M})$ when $\mathcal{H}$ is finite-dimensional. Theorem 1 together with Theorem 2 show that our approach can provably train a well-performed job selector, even when the unseen job exists and raw data are unavailable.

6 Related Work and Discussion

In this section, we discuss related works from two aspects.

Related Topics. Domain adaptation (Ben-David et al. 2006) and transfer learning (Pan and Yang 2010) aim to adapt source data to help the training on target data. The problem is that the raw source data are available when training the target model, which is not applicable in the learnware scenario. More relevant topics to ours are learning from auxiliary classifiers (Duan et al. 2009) or hypothesis transfer learning (Kuzborskij and Orabona 2013, 2017; Zhao, Cai, and Zhou 2020), where researchers attempt to exploit pre-trained models for handling learners' current jobs. Their assumption is that the given pre-trained models are always helpful for the current job, which exhibits striking difference from our problem as helpful models could be only a few even non-existent in the learnware market. Multi-party learning (Pathak, Rane, and Raj 2010; Wu, Liu, and Zhou 2019) also considers uniting local data to solve the same/similar job in privacy-preserving ways. But they assume that every local data is relevant to the user's current job, which is not the case in learnware as the reusability is one of its main concerns.

There are recent efforts devoted to the issue of reusability. Specifically, Wu et al. (2020) make the first attempt to reuse a pool of model with the RKME specification. Later, Ding and Zhou (2020) use outlier detectors as the specification and develop a boosting-based approach to deploy models. The difference between our work and previous studies is that we consider the existence of the unseen job, the identification of which requires new deploying algorithms.



(a) Current job (b) Predictions

Figure 3: Deploying phase

Related Techniques. The main technical stuffs, mixture proportion estimation (Jain et al. 2016; du Plessis, Niu, and Sugiyama 2017; Bekker and Davis 2018) and risk rewriting technique (du Plessis, Niu, and Sugiyama 2014, 2015; Zhang et al. 2020), have been developed in the research field of weakly supervised learning. However, all the existing methods rely on the raw data, which are not applicable in the learnware scenario. Our contribution is to modify and integrate them with the RKME specification, where we theoretically justify the compatibility of our modification.

7 Experiments

This section examines the efficacy of our method, where we compare our method with contenders in various scenarios. In the following, we first illustrate our method by a toy example, followed by the comparisons on benchmark datasets.

7.1 Toy Example

Following the empirical studies of Wu et al. (2020), we first illustrate the RKME based learnware by a toy classification problem. There are two stages of the toy experiment.

Uploading Phase. Figure 2 illustrates the uploading phase. There are three developers in the market, whose local datasets are denoted by "△", "○", "□". Each local dataset contains 160 samples and is generated from mixture of different Gaussian distributions. Labels of each dataset are decided by a circle where yellow ones are positive ("$+$") while blue ones are negative ("$-$"). Based on their training data, developers construct learnwares individually, where linear models are trained over the local datasets and RKMEs are built. The reduced set size is $M = 5$, equaling $\lfloor \ln 160 \rfloor$ and is quite fewer comparing with the raw training data.

Deploying Phase. Figure 3 illustrates the deploying phase, where the current job is taken as a mixture of uploaded jobs and an unseen part (denoted by "☆"). The true mixture proportion of the uploaded jobs is $[0.5, 0.2, 0, 0.3]$ for "△", "○", "□" and "☆". By exploiting the unlabeled testing data and RKMEs of the uploaded jobs, the estimated proportions returned by our RKME based MPE estimator are $[0.506, 0.182, 0.013, 0.299]$, which is close to the ground truth. Figure 3(b) shows the the decision boundary of the selector, which essentially identifies the unseen job samples and assign the rest to proper models.

Datasets	# Job	Instance-recurrent assumption (2)		Unseen-job assumption (3)				
		RKME-basic	Ours	RKME-OCSVM	RKME-iForest	Ours	Ours-oracle	Oracle
CIFAR-100	2	75.90 ± 5.41	**79.28 ± 4.84**	62.81 ± 6.13	58.55 ± 4.74	**90.32 ± 2.02**	90.39 ± 2.22	93.55 ± 1.52
	5	**75.43 ± 3.92**	72.59 ± 4.80	49.80 ± 2.73	37.85 ± 2.64	**76.04 ± 4.84**	79.15 ± 2.01	87.88 ± 2.77
	10	**75.95 ± 2.34**	73.28 ± 2.10	44.70 ± 2.02	29.18 ± 2.97	**72.42 ± 3.81**	74.49 ± 2.71	86.43 ± 1.95
Newsgroup20	2	85.75 ± 7.59	84.31 ± 8.34	56.44 ± 7.03	59.01 ± 5.46	**79.91 ± 9.98**	88.73 ± 7.48	93.64 ± 4.94
	3	88.18 ± 6.69	86.70 ± 6.79	52.93 ± 3.81	48.58 ± 3.59	**75.56 ± 6.39**	83.43 ± 4.21	90.55 ± 3.61
	4	**87.10 ± 6.38**	83.32 ± 6.89	48.94 ± 2.69	43.75 ± 3.53	**71.19 ± 6.15**	80.90 ± 2.50	89.88 ± 2.09
ELT Character	2	87.47 ± 5.04	88.15 ± 4.26	31.56 ± 7.56	38.15 ± 12.8	**91.85 ± 2.63**	93.14 ± 2.06	95.67 ± 2.74
	3	**87.09 ± 2.35**	84.15 ± 3.28	37.79 ± 9.05	37.27 ± 10.0	**85.52 ± 5.30**	88.85 ± 4.40	95.49 ± 1.61
	4	**86.23 ± 2.63**	81.45 ± 2.93	44.75 ± 4.59	39.49 ± 6.55	**76.01 ± 5.53**	84.61 ± 2.97	94.10 ± 1.67

Table 1: Accuracy on true labels. The best *feasible* method is emphasized in bold (paired t-tests at 5% significance level).

7.2 Benchmark Datasets

We evaluate our method on benchmark datasets to show its effectiveness on reusing models. As the reusability of a pool of models is a new problem, we first compare with the baseline.

- *RKME-basic* (Wu et al. 2020) exploits the RKME to reuse the model pool under the instance-recurrent assumption (2), where the existence of unseen job is not considered. We take this method as a baseline.

To evaluate how well our method can perform, we compare with two skylines, which is not feasible in real applications.

- *Oracle* knows the job each sample belongs to. It enjoys the best performance that the pool of models can achieve.
- *Ours-oracle* uses our method to train the selector with true $\{w_i\}_{i=1}^C$, served a skyline for the MPE estimator.

Meanwhile, since *RKME-basic* method cannot identify the unseen job, we equip it with novelty detectors, which can be seen as another specification provided by the developer.

- *RKME-OCSVM* equips *RKME-basic* with the one-class SVM (Schölkopf et al. 2001), where an OCSVM is provided for each uploaded job as an additional specification. When a sample is rejected by all one-class SVMs of the uploaded jobs, it is predicted as from the unseen job. The rest samples are predicted with *RKME-basic*.
- *RKME-iForest* equips *RKME-basic* with iForest (Liu, Ting, and Zhou 2008) to predict the unseen job.

For all RKHS based methods, we exploit the Gaussian kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma\|\mathbf{x} - \mathbf{y}\|_2^2)$ with $\gamma = 0.01$. The reduced set size is $M = 10$ which is very tiny size. As for our method, we set $\nu = 0.25$ for the MPE estimator and choose the square loss $\psi(z) = (1 - z)^2/4$ for the job selector.

Datasets. The evaluation is conducted on three widely used benchmark datasets: CIFAR-100 (Krizhevsky 2009), Newsgroup20 (Joachims 1997) and the ETL character dataset. CIFAR-100 has 100 classes and are naturally divided into 20 parts with 5 classes in each. We take each part as a job, which contains 2500 training instances. Newsgroup20 has a similar configuration, where there are 5 jobs and each has 2000 instances. The character dataset consists of characters from 6 different scripts. The classification of each script is seen as a job and has at least 6528 instances.

Configuration. To evaluate the method, we compare it with contenders in various scenarios. First, we compare to *RKME-basic* under the instance-recurrent assumption (2), where no unseen job appears. *RKME-basic* is specifically designed for this scenario. The goal of the comparison is to evaluate the safety of our method even if it always assumes the existence of potential unseen jobs. Next, we conduct experiments with emerging unseen jobs, where only parts of local jobs are uploaded, and there is always one unseen job. In both scenarios, the current job is stimulated by randomly mixing a different number of local jobs with equal mixture proportions. All experiments are repeated 10 times and more experimental details can be founded in Appendix C.

Results. Table 1 presents the prediction accuracy over the *true label* of the testing data, where the unseen job is assumed to have a unique class. We provide more evaluation on other measures in Appendix C. As shown in the comparison under the instance-recurrent assumption (2), our method achieves comparable performance with *RKME-basic* in various numbers of mixed jobs, which shows that our method is safe to use when there is no unseen job. The comparison under the unseen-job assumption (3) validates the superiority of our method. The novel detectors do not achieve the expected performance. The reason might be that it is solely trained over the local dataset, whose predictions on other jobs might not be accurate. On the contrary, by exploiting the unlabeled testing data, our method can align binary classifiers in the OVR scheme to select the job a testing sample comes from.

Our method achieves comparable performance with *Ours-oracle*, which shows that the MPE estimator is accurate since *Ours-oracle* is trained with the true $\{w_i\}_{i=1}^C$. The comparison between *Ours-oracle* and *Oracle* further validates our job selector's efficacy, as *Ours-oracle* trains the selector with only a tiny reduced set size $M = 10$ while *Oracle* is assumed to know each sample's ground truth job.

8 Conclusion

This paper attempts to address the challenge of how to reuse learnwares when the user's job involves unseen parts not covered by the current learnware market. Based on the recently proposed RKME specification and advances in weakly supervised learning, we design a provably effective approach that can identify the unseen job samples while assigning the rest to proper models. Empirical evaluation validates that our approach can be safely applied no matter whether there are unseen jobs or not. An interesting future issue is to consider the labeling cost budget.

References

Bach, F. R.; Lacoste-Julien, S.; and Obozinski, G. 2012. On the equivalence between herding and conditional gradient algorithms. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 1355–1362.

Bekker, J.; and Davis, J. 2018. Estimating the class prior in positive and unlabeled data through decision tree induction. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI)*, 2712–2719.

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2006. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 19 (NeurIPS)*, 137–144.

Blanchard, G.; Lee, G.; and Scott, C. 2010. Semi-supervised novelty detection. *Journal of Machine Learning Research* 11: 2973–3009.

Burges, C. J. C. 1996. Simplified support vector decision rules. In *Proceedings of the 13th International Conference on Machine Learning (ICML)*, 71–77.

Chen, Y.; Welling, M.; and Smola, A. J. 2010. Super-samples from kernel herding. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, 109–116.

Ding, Y.-X.; and Zhou, Z.-H. 2020. Boosting-based reliable model reuse. In *Proceedings of the 12th Asian Conference on Machine Learning (ACML)*, 145–160.

Doran, G. 2013. Distribution kernel methods for multiple-instance learning. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, 1660–1661.

du Plessis, M. C.; Niu, G.; and Sugiyama, M. 2014. Analysis of learning from positive and unlabeled data. In *Advances in Neural Information Processing Systems 27 (NeurIPS)*, 703–711.

du Plessis, M. C.; Niu, G.; and Sugiyama, M. 2015. Convex formulation for learning from positive and unlabeled Data. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 1386–1394.

du Plessis, M. C.; Niu, G.; and Sugiyama, M. 2017. Class-prior estimation for learning from positive and unlabeled data. *Machine Learning* 463–492.

Duan, L.; Tsang, I. W.; Xu, D.; and Chua, T. 2009. Domain adaptation from multiple sources via auxiliary classifiers. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, 289–296.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. J. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13: 723–773.

Jain, S.; White, M.; Trosset, M. W.; and Radivojac, P. 2016. Nonparametric semi-supervised learning of class proportions arXiv:1601.01944.

Joachims, T. 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the 14th International Conference on Machine Learning (ICML)*, 143–151.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.

Kuzborskij, I.; and Orabona, F. 2013. Stability and hypothesis transfer learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 942–950.

Kuzborskij, I.; and Orabona, F. 2017. Fast rates by transferring from auxiliary hypotheses. *Machine Learning* 106(2): 171–195.

Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation forest. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM)*, 413–422.

Muandet, K.; Fukumizu, K.; Sriperumbudur, B. K.; and Schölkopf, B. 2017. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends in Machine Learning* 10(1-2): 1–141.

Muandet, K.; and Schölkopf, B. 2013. One-class support measure machines for group anomaly detection. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, 449–458.

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

Pathak, M. A.; Rane, S.; and Raj, B. 2010. Multiparty differential privacy via aggregation of locally trained classifiers. In *Advances in Neural Information Processing Systems 23 (NeurIPS)*, 1876–1884.

Ramaswamy, H. G.; Scott, C.; and Tewari, A. 2016. Mixture proportion estimation via kernel embeddings of distributions. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2052–2060.

Schölkopf, B.; Mika, S.; Burges, C. J. C.; Knirsch, P.; Müller, K.; Rätsch, G.; and Smola, A. J. 1999. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks* 10(5): 1000–1017.

Schölkopf, B.; Platt, J. C.; Shawe-Taylor, J.; Smola, A. J.; and Williamson, R. C. 2001. Estimating the support of a high-dimensional distribution. *Neural Computation* 13(7): 1443–1471.

Schölkopf, B.; and Smola, A. J. 2002. *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* Adaptive Computation and Machine Learning Series. The MIT Press.

Scott, C. 2015. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 838–846.

Smola, A. J.; Gretton, A.; Song, L.; and Schölkopf, B. 2007. A Hilbert space embedding for distributions. In *Proceedings of the 18th International Conference on Algorithmic Learning Theory (ALT)*, 13–31.

Sriperumbudur, B. K.; Fukumizu, K.; and Lanckriet, G. R. G. 2011. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research* 12: 2389–2410.

Wu, X.-Z.; Liu, S.; and Zhou, Z. 2019. Heterogeneous model reuse via optimizing multiparty multiclass margin. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 6840–6849.

Wu, X.-Z.; Xu, W.; Liu, S.; and Zhou, Z.-H. 2020. Model reuse with reduced kernel mean embedding specification arXiv:2001.07135.

Zhang, T. 2004. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research* 5: 1225–1251.

Zhang, Y.-J.; Zhao, P.; Ma, L.; and Zhou, Z.-H. 2020. An unbiased risk estimator for learning with augmented classes. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 10247–10258.

Zhao, P.; Cai, L.-W.; and Zhou, Z.-H. 2020. Handling concept drift via model reuse. *Machine Learning* 109(3): 533–568.

Zhou, Z.-H. 2016. Learnware: On the future of machine learning. *Frontiers of Computer Science* 10(4): 589–590.

Zhou, Z.-H. 2018. A brief introduction to weakly supervised learning. *National Science Review* 5(1): 44–53.