Rectify Heterogeneous Models with Semantic Mapping

Han-Jia Ye [1] De-Chuan Zhan [1] Yuan Jiang [1] Zhi-Hua Zhou [1]

Abstract

On the way to the robust learner for real-world applications, there are still great challenges, including considering unknown environments with limited data. *Learnware* (Zhou, 2016) describes a novel perspective, and claims that learning models should have *reusable* and *evolvable* properties. We propose to Encode Meta InformaTion of features (EMIT), as the model specification for characterizing the changes, which grants the model evolvability to bridge heterogeneous feature spaces. Then, pre-trained models from related tasks can be Reused by our REctiFy via heterOgeneous pRedictor Mapping (REFORM) framework. In summary, the pre-trained model is adapted to a new environment with different features, through model refining on only a small amount of training data in the current task. Experimental results over both synthetic and real-world tasks with diverse feature configurations validate the effectiveness and practical utility of the proposed framework.

1. Introduction

As machine learning has been successfully applied in many real-world applications, the robustness of the learner is attracting more attention (Dietterich, 2017). Increasing the robustness of models in dynamic environments is desirable in real-world scenarios. For example, dictionaries encode words for documents classification, whose keys change as hot topics appear/vanish with time; In a recommendation system, statistics on interactions over items are characterized as user profiles, which fluctuates with newly arrival and out-dated items; Although targeting the same goal, branches of a company deal with locality specific features apart from the general ones, which hampers the experience exchange between branches. In summary, it is the *feature set transition* that becomes one of the fundamental issues in a non-stationary environment, as in Fig. 1. Besides, due to the expensive labeling cost, there are usually only *a few collected examples* such as newly labeled documents for new circumstances later, especially within a short period.

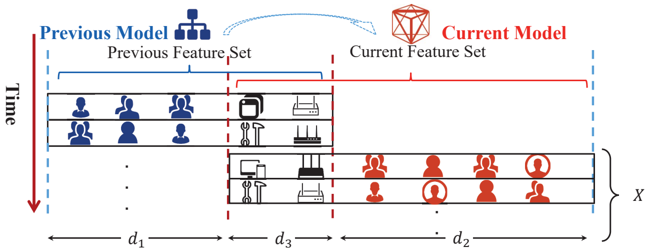


Figure 1. Example of modeling with heterogeneous feature spaces as environment changing. The number/types of extracted features for each instance (each row) will increase or decrease. The two task specific and shared feature dimensions are d_1, d_2, and d_3, respectively. To increase the robustness of models, the goal of this paper is to smartly utilize limited current task data X and the previous *well-trained* model (over $d_1 + d_3$ features) to improve the performance of the current task (with $d_3 + d_2$ dimensions).

Learnware (Zhou, 2016) describes a novel perspective towards the robust modeling, which is a well-performed pre-trained learner with specifications. Two essential properties of *learnware*, i.e., the *reusable* and *evolvable*, are emphasized in this work. Specifically, reusability ensures that for a new related target, the model is capable of being enhanced, adapted, and refined easily, with only limited new task data. Evolvability considers the non-stationary nature of the environment, so that the model is able to handle variations in the environment, ensuring that it can be reused for tasks with heterogeneous feature spaces.

This paper makes a preliminary step towards robust modeling guided by *learnware*, containing two parts implementing the *reusable* and *evolvable* properties accordingly. We develop a new model reuse framework on *heterogeneous* feature spaces in a dynamic environment, and propose a novel evolvability solution via linking different feature spaces.

Popular approaches upon landmark (Gong et al., 2013), instance weights (Sugiyama & Kawanabe, 2012), or sub-

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China. Correspondence to: De-Chuan Zhan <zhandc@lamda.nju.edu.cn>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

space (Bhattarai et al., 2016) require former task data to determine the task relevance, whose models cannot be directly reused in varying environments. In contrast, our framework REctiFy via heterOgeneous pRedictor Mapping (REFORM) utilizes the *well-trained model* from past environment effectively, even with diverse features. It is the inconsistency between heterogeneous features that impedes the application of the old model. If the features correspondence across tasks is known in advance, REFORM bridges this heterogeneity gap with a semantic mapping by optimal transport (Villani, 2008). Otherwise, we propose a novel strategy named Encode Meta InformaTion of features (EMIT), discovering the meta feature representations by dictionary reconstruction. EMIT leverages a wide range of related tasks, and aims at revealing the invariant regularities over features as task shifting. It makes REFORM different from homogeneous domain transferring (Long et al., 2014) or cross-modal adaptation with paired examples (Kulis et al., 2011).

Therefore, after EMIT offers the feature correspondence with meta encoding, REFORM refines a model from heterogeneous feature space through only a small amount of new task training data. Two implementations of REFORM are investigated on both synthetic and real-world tasks under varying environments. Experiments validate the superiority of REFORM, and its possession of *learnware*'s properties.

We start with theoretical intuitions on model reuse and then describe the REFORM framework, including the EMIT strategy and two concrete implementations. Next is related literature followed by experiments and the conclusion.

2. Notations

Consider a C-class classification task with data $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$, $\|\mathbf{x}_i\|_2 \le \chi$, and $\mathbf{y}_i \in \{-1, 1\}^C$. The position of 1 in $\mathbf{y}_i$ indicates the class of $\mathbf{x}_i$. Every example $(\mathbf{x}_i, \mathbf{y}_i)$ is drawn from $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, with $\mathcal{X}$ and $\mathcal{Y}$ corresponding to the instance and label distributions. $\operatorname{diag}(\cdot)$ transforms the input vector to a diagonal matrix. $\Delta_d = \{\boldsymbol{\mu} : \boldsymbol{\mu} \in \mathbb{R}_*^d, \boldsymbol{\mu}^\top \mathbf{1} = 1\}$ denotes the set of d-dimensional simplex. $\mathbf{1}$ is a vector with all elements equal to 1, whose size can be determined from the context.

3. Model Reuse and REFORM

This section starts with a theoretical explanation on how to take advantage of a related homogeneous model and limited data in the current task. Based on this, we describe the main idea of the REctiFy via heterOgeneous pRedictor Mapping (REFORM) framework, building a semantic map to reuse model from heterogeneous feature spaces. Then we present the key component EMIT for feature meta information encoding/management, which endows the framework handling changed features in the dynamic environment.

3.1. Model Reuse on Homogeneous Features

Consider a linear classifier $f(\mathbf{x}_i) = W^\top \mathbf{x}_i \in \mathbb{R}^C$ predicts over the centralized instance $\mathbf{x}_i$. Model $W \in \mathbb{R}^{d \times C}$, with columns corresponding to each class, can be learned by:

$$\min_W \frac{1}{N} \sum_{i=1}^N \ell(f(\mathbf{x}_i) - \mathbf{y}_i) + \lambda \|W\|_F^2 \ .$$

Loss function $\ell(\cdot) : \mathbb{R}^C \to \mathbb{R}_*$ measures the difference between *vector form* class affiliation prediction and the true label, the smaller the better. Instead of learning the linear predictor W directly, in the model reuse scenario, the helpfulness of the model $W_0 \in \mathbb{R}^{d \times C}$ from a related task is stressed, which gives rise to the target function:

$$\underbrace{\min_W \frac{1}{N} \sum_{i=1}^N \ell(f(\mathbf{x}_i) - \mathbf{y}_i)}_{\text{Empirical Risk } \epsilon_N(W)} + \lambda \|W - W_0\|_F^2 \ . \tag{1}$$

$\epsilon_N(W)$ depends on the N examples of the current task. Instead of optimizing empirical loss directly, Eq. 1 reuses previous model W_0 as a biased regularizer, which ensures the current model W will not deviate far away from the provided W_0. Learning by Eq. 1 can also be transformed to learn a model bias ΔW based on the existing W_0, and then predicts with $W_0 + \Delta W$ (Tommasi et al., 2014). The expected risk of $\epsilon_N(W)$ is $\epsilon(W) = \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim \mathcal{Z}}[\ell(f(\mathbf{x}) - \mathbf{y})]$. We prove that the consideration of a well-trained model from a related homogeneous task facilitates the learning efficiency in the current multi-class task, i.e., the convergence rate from $\epsilon_N(W)$ to $\epsilon(W)$ is influenced by W_0.

Theorem 1 *Consider a C-class learning problem over $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ as in Eq. 1, which has a M-bounded L-Lipschitz vector valued loss function w.r.t. Euclidean norm. Define $\mathcal{W} = \{W : \|W - W_0\|_F \le \sqrt{\frac{\epsilon_N(W_0)}{\lambda}}, \epsilon_N(W) \le \epsilon_N(W_0)\}$. Set $C_1 = (\frac{2}{3} + 4LC\chi)M \log 1/\delta$, and $C_2 = \frac{4LC\chi+2}{\sqrt{\lambda}} + \sqrt{2M \log 1/\delta}$, then for every $W \in \mathcal{W}$ and $0 < \delta < 1$, with probability at least $1 - \delta$, we have:*[1]

$$\epsilon(W) \le \epsilon_N(W) + \frac{C_1}{N} + C_2 \sqrt{\frac{\epsilon(W_0)}{N}} \ . \tag{2}$$

Theorem 1 provides a $\mathcal{O}(\frac{1}{\sqrt{N}})$ convergence rate for the generalization error when learning the model W, which is consistent with (Bartlett & Mendelson, 2002; Maurer, 2016). This convergence rate is directly related to the sample complexity, i.e., the faster the rate, the smaller the number of training examples is required to obtain a certain risk difference. If the provided model W_0 adapts well on the current task distribution, i.e., a small expected risk $\epsilon(W_0) \to 0$, such that the r.h.s. of Eq. 2 will be more compact and

[1] The detailed proof can be found in the supplementary material (http://lamda.nju.edu.cn/yehj/reform-supp.pdf).

achieves a faster rate whose order is $\mathcal{O}(\frac{1}{N})$. Here $\epsilon(W_0)$ naturally acts as a task relatedness measure. Thus, with an uninformative prior Eq. 1 converges in a general rate; but reusing suitable *related* model helps reduce the sample complexity for target learning problem, and can even improve the order of learning rate. In other words, with *limited current task examples*, the current learned model W can achieve higher performance in expectation.

3.2. Reuse Heterogeneous Feature Space Model

The above analysis is limited to the case reusing a well-trained model within the same feature space. However, real-world environment is not stationary, and the transition between feature sets limits the direct model reuse between feature domains. We extend the model reuse to the heterogeneous case by *constructing a semantic map between feature sets as well as models*, which enables the current task to leverage related heterogeneous models.

Considering that the variant feature spaces across tasks can be substantially related, and model reuse on heterogeneous feature spaces should focus on the *feature mapping* between original and later features sets. If each feature has a corresponding probability distribution, the map can be obtained by the coupling between their normalized marginal probability mass vectors $\boldsymbol{\mu}_1 \in \Delta_{d_1}$ and $\boldsymbol{\mu}_2 \in \Delta_{d_2}$. For the practicability and comprehensibility, we introduce a matrix $Q \in \mathbb{R}_*^{d_2 \times d_1}$ to depict the feature variation relationship, i.e., the semantic cost changing features from current to former task. Thus, the feature space map $T \in \mathbb{R}_*^{d_2 \times d_1}$ can be obtained by minimizing the total transportation cost:

$$\min_T \langle T, Q\rangle \quad s.t.\ T\mathbf{1} = \boldsymbol{\mu}_2, T^\top \mathbf{1} = \boldsymbol{\mu}_1, T \geq 0\ . \qquad (3)$$

Eq. 3 is also the Kantorovitch formulation of the Optimal Transport (OT) problem (Villani, 2008), which aligns two distributions by the learned coupling T. So T shows how to do a semantic map from one set to another. The probability mass of a feature will be moved to similar ones, i.e., those features with small costs. This feature semantic map can also be applied *on the model space*, i.e., coefficients in one model can be transported to another weighted by their feature similarity. For example, in a simple case when we exchange positions of features to construct a new feature space, the cost matrix Q will be formed as a square permutation like matrix revealing the correspondence between features. With uniform feature marginal, OT will output a permutation matrix with the right alignment between two feature sets (Courty et al., 2017b). Applying this alignment of features over models, we can transform a "well-trained" classifier from former task to the current one perfectly. In a general scenario, transforming model based on feature transportation plan is also meaningful, since model coefficients for similar features usually have similar values. For instance, when each feature represents a word, and cost depicts their physical similarities, then predictor weights for "Trump" maybe close to "Obama" (Kusner et al., 2015).

We propose our REctiFy via heterOgeneous pRedictor Mapping (ReForm) framework, reusing the model from related task even the feature space changes. In detail, for current task with dimension $d = d_2$, the goal is to reuse a well-trained model $\hat{W}_0 \in \mathbb{R}^{d_1 \times C}$ from related task with dimension d_1. The main ReForm idea is to utilize the semantic map $T \in \mathbb{R}_*^{d_2 \times d_1}$ between two feature spaces to link models by setting the prior $W_0 = d_2 T \hat{W}_0$. d_2 in the transformation scales the marginal probability. Based on this, we refine W_0 with limited examples from the current task as in Eq. 1.

3.3. Cost Matrix and Meta Feature Representation

It is obvious that the cost matrix Q fully characterizes the influence of the environmental change, i.e., the relationship between heterogeneous feature spaces. Sometimes it can be provided manually by measuring physical similarities between two features. To make the model evolvable, we propose to generate Q based on feature meta representations, which can be easily collected in real-world tasks. For example, each word in the task-specific dictionary can be represented in a word2vec (Mikolov et al., 2013) way. Benefiting from the invariant nature of feature meta representations, utilizing which as the model specification depicts the evolvable property over the environment, and facilitates the construction of feature relationship, especially in the non-stationary environment with different features. Therefore, Q can be computed as the pairwise (squared) Euclidean distance between corresponding feature meta vectors.

The ReForm framework can also be explained from a reconstruction perspective in the *feature meta space*. Given meta sets $M_f = \{\mathbf{m}_m\}_{m=1}^{d_1} \in \mathbb{R}^{D \times d_1}$ and $M = \{\mathbf{m}_n\}_{n=1}^{d_2} \in \mathbb{R}^{D \times d_2}$, each column is a D dimension meta representation of features in former and current tasks. Although models have different feature dimensions, d_1 and d_2, we focus on their common regularities, i.e., feature meta space where all features are in the *same* representation form. Therefore, we analyze the change of features in this meta space, and attribute the feature change to the distribution variation between meta representations in this space. Then relationship between two sets of meta representation can be discovered by OT as in Eq. 3 (Courty et al., 2017b), and the learned coupling $T \in \mathbb{R}_*^{d_2 \times d_1}$ directs how to transport one set of meta feature to another with the lowest cost: given T, a particular meta feature $\mathbf{m}_n$ will be transferred to $\hat{\mathbf{m}}_n$ in the domain of M_f by (Perrot et al., 2016):

$$\hat{\mathbf{m}}_n = \arg\min_{\mathbf{m}} \sum_{m=1}^{d_1} T_{n,m} \|\mathbf{m} - \mathbf{m}_m\|^2\ ,\ n = 1, \ldots, d_2\ . \qquad (4)$$

Optimization in Eq. 4 has a closed form solution that $\hat{M} = M_f(\operatorname{diag}(T\mathbf{1})^{-1}T)^\top$, which can be further simplified to $\hat{M} = d_2 M_f T^\top$ if we assume the marginal distribution is uniform. This transformation can be thought as using the coefficient $d_2 T^\top$ to reconstruct meta features of domain M us-

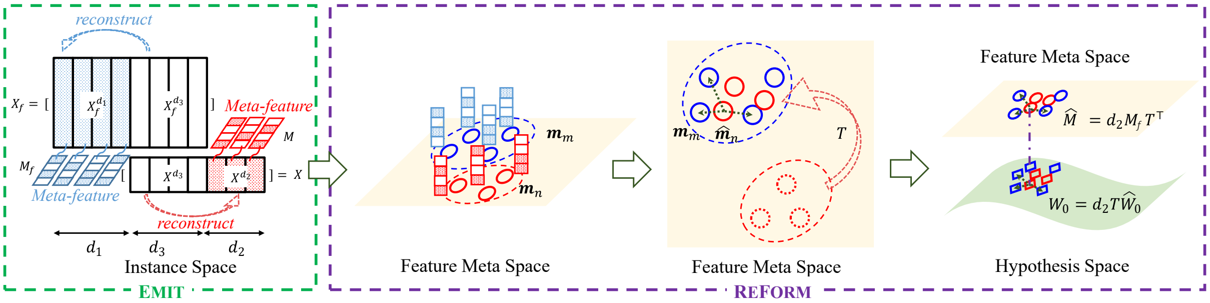


Figure 2. Illustration of the EMIT and REFORM flows. If no semantic embeddings provided, feature meta representation could be constructed in a reconstruction manner as in the left plot. In the feature meta space (right plots), meta representations of features build the transportation cost, and the corresponding relationship between features could be discovered by optimal transport in this space (with uniform marginals). The reconstruction coefficients of new features by the old ones also apply to the reconstruction relationship between two domain-specific models. It is expected that the transformed model can be easily adapted to the current task.

ing the meta features from domain M_f. REFORM assumes this relationship also apply to *the model space*, where we “reconstruct” the classifier W_0 w.r.t. meta domain M using model $\hat{W}_0$ w.r.t. domain M_f by $W_0^\top = \hat{W}_0^\top (d_2 T)^\top$. This process keeping reconstruction relationship across feature and model spaces is illustrated in Fig. 2, where REFORM deals with heterogeneous feature spaces by $W_0 = d_2 T \hat{W}_0$.

3.4. EMIT: Encoding Feature Meta Information

In the scenario which is hard to obtain the concrete meaning of features or no provided meta information, we propose a novel strategy, Encode Meta InformaTion of features (EMIT), to enable learning in the REFORM way. We focus on the case that former and current tasks *have shared features*. To get the same form of meta representation of features for two tasks, EMIT operates by reconstructing task-specific features with dictionaries, i.e., connecting two non-overlapping task features using their shared part. We decompose former task features (with N_f instances) X_f and current features X as $X_f = [X_f^{d_1} \in \mathbb{R}^{N_f \times d_1}, X_f^{d_3} \in \mathbb{R}^{N_f \times d_3}]$ and $X = [X^{d_3} \in \mathbb{R}^{N \times d_3}, X^{d_2} \in \mathbb{R}^{N \times d_2}]$. Since components $X_f^{d_3}$ and X^{d_3} correspond to task shared features and have the same feature meaning, we can use them to represent/reconstruct $X_f^{d_1}$ and X^{d_2}, respectively:

$$\|X_f^{d_1} - X_f^{d_3} M_f\|_F^2 + \lambda \sum_{m=1}^{d_1} \|M_{f,m}\|_0 \ , \tag{5}$$

$$\|X^{d_2} - X^{d_3} M\|_F^2 + \lambda \sum_{n=1}^{d_2} \|M_n\|_0 \ . \tag{6}$$

$M_f \in \mathbb{R}^{d_3 \times d_1}$ and $M \in \mathbb{R}^{d_3 \times d_2}$ are reconstruction coefficients, whose m-th and n-th columns $M_{f,m} \in \mathbb{R}^{d_3}$ and $M_n \in \mathbb{R}^{d_3}$ correspond to the coefficients for particular features, and can be used as feature meta representations. $\lambda > 0$ is the regularization parameter, which controls the sparsity of reconstruction results. Eq. 5 and Eq. 6 obtain the same form reconstruction coefficients by using corresponding same meaning parts $X_f^{d_3}$ and X^{d_3} as dictionaries, which can be solved by Orthogonal Matching Pursuit (OMP) efficiently. Thus, for two overlapping feature sets, we get M_f and M first, then the feature transition cost matrix Q can be computed by their pairwise (squared) Euclidean distance. It is noteworthy that the EMIT is *unsupervised*, which can incorporate unlabeled data and get better reconstructions.

With EMIT, meta representations can be constructed independently during the training process of the former task. The pass of model and reconstruction coefficients keeps the raw data privacy during the model reuse. Besides, feature meta information helps the model perceive the change of the environment, i.e., variations on features. Thus, EMIT endows the evolvability of a model even in heterogeneous spaces and acts as a key step in the REFORM framework. More discussions on REFORM and EMIT are in the supp.

4. Framework Implementations

The REFORM framework points out a general way to reuse related model from tasks with heterogeneous features. Since constructing the semantic map with the optimal transport process does not take current task examples into consideration, hence directly learning with the help of prior W_0 by Eq. 1 still has some drawbacks. We focus on the transition between the *non-overlapping* parts between two feature spaces and implement two variants of our REFORM framework. First, an adaptive scale approach is designed, then the map optimization is incorporated in current task training.

Assume former task specific features (d_1-dimension) come first, task shared features (d_3-dimension) in the second, and current task specific features (d_2-dimension) at last, as shown in Fig. 1. The well-trained former task model can be decomposed into two parts, $\hat{W}_0 = [\hat{W}_0^{d_1}; \hat{W}_0^{d_3}]$, according to the task specific and shared dimensions, i.e., $\hat{W}_0^{d_1} \in \mathbb{R}^{d_1 \times C}$ and $\hat{W}_0^{d_3} \in \mathbb{R}^{d_3 \times C}$. Similarly, for current task classifier, we have $W = [W^{d_3}; W^{d_2}]$, and the transformed prior $W_0 = [W_0^{d_3}; W_0^{d_2}]$. The goal of ReForm implementation is to reuse $\hat{W}_0$ the from previous task in the current learning process of W, and improve the current performance with limited training examples (X, Y).

4.1. Implementation with Adaptive Scale

The original form of the optimal transported model $W_0^{d_2} = d_2 T \hat{W}_0^{d_1}$ lacks the flexibility over features with different scales and complex mapping relationships. On the one hand, direct scale by d_2 may be insufficient; on the other hand, new features will have negative relationships with old ones, or there may exist redundant mapping between features. Thus, we decompose the scale and model part of a classifier. With $W_0^{d_2}$ serving as the model part, we add a class-specific scale matrix $A \in \mathbb{R}^{d_2 \times C}$ to take scale and sign into consideration, which results in $W_0 = [\hat{W}_0^{d_3}; A \odot d_2 T \hat{W}_0^{d_1}] = [\hat{W}_0^{d_3}; A \odot W_0^{d_2}]$. Notation $\odot$ denotes the element-wise product. Therefore, current classifier W and scale coefficients A can be learned in the objective jointly:

$$\min_{W, A, \mathbf{b}} \|XW + \mathbf{1}\mathbf{b}^\top - Y\|_F^2 + \lambda_1 \|W - W_0\|_F^2 + \lambda_2 \|A\|_F^2 \; . \quad (7)$$

The first two terms in Eq. 7 learn a classifier like least square SVM (Ye & Xiong, 2007), but biased w.r.t. the transformed model W_0. The third term tunes the scale and sign of transformed classifier. $\mathbf{b} \in \mathbb{R}^C$ is a bias vector, and λ_1, λ_2 are non-negative parameters. Using the fact that $\mathbf{b} = \frac{1}{N}(Y^\top \mathbf{1} - W^\top X^\top \mathbf{1})$, we can introduce the centralization matrix $H = I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$, and get rid of the bias vector $\mathbf{b}$ as:

$$\min_{W, A} \|HXW - HY\|_F^2 + \lambda_1 \|W - W_0\|_F^2 + \lambda_2 \|A\|_F^2 \; , \quad (8)$$

then the problem can be solved in an alternative manner. With fixed scale matrix, we reuse former task transformed classifier W_0 to help the learning of current model; while for a particular classifier, the scale of the related model is tuned based on the current training data. The scaled matrix A is initialized that all values are equal to one first, then the current task classifier W can be solved in the closed form:

$$W = (X^\top H X + \lambda_1 I)^{-1}(\lambda_1 W_0 + X^\top H Y) \; . \quad (9)$$

This solution can be simplified in high dimensional case with Woodbury identity. To deal with the scale matrix, we first reformulate the optimization problem as:

$$\min_{A} \lambda_1 \|W^{d_2} - A \odot W_0^{d_2}\|_F^2 + \lambda_2 \|A\|_F^2 \; ,$$

and then decompose the sub-problem for each class separately. For the c-th class we have $\min_{\mathbf{a}_c} \lambda_1 \|W_c^{d_2} - \mathbf{a}_c \odot W_{0,c}^{d_2}\|_F^2 + \lambda_2 \|\mathbf{a}_c\|_F^2 = \min_{\mathbf{a}_c} \lambda_1 \|W_c^{d_2} - \mathrm{diag}(W_{0,c}^{d_2})\mathbf{a}_c\|_F^2 + \lambda_2 \|\mathbf{a}_c\|_F^2$. $\mathbf{a}_c$, $W_c^{d_2}$, $W_{0,c}^{d_2}$ are the c-th column of matrix A, W^{d_2}, and $W_0^{d_2}$, respectively. Then, we can also get $\mathbf{a}_c = (\lambda_1 \mathrm{diag}(W_{0,c}^{d_2} \odot W_{0,c}^{d_2}) + \lambda_2 I)^{-1} \lambda_1 (W_{0,c}^{d_2} \odot W_c^{d_2})$ in a closed form. In summary, when reusing a related heterogeneous model from the previous task, this ReForm implementation learns a classifier scale by taking advantage of current task data, which is able to consider the negative transformation relationship and identify redundant maps.

4.2. Implementation with Learned Transformation

To fully utilize data in the current task, the ReForm implementation can also incorporate the optimal transportation process during training to find a semantic map with the current data, which is different from the previous approach using a pre-computed transportation plan. The target is

$$\begin{aligned}
&\min_{W, \mathbf{b}, T} \|Y - XW - \mathbf{1}\mathbf{b}^\top\|_F^2 + \lambda_1 \|W - W_0\|_F^2 + \lambda_2 \langle T, Q \rangle \\
&s.t. \;\; W_0 = [\hat{W}_0^{d_3}; d_2 T \hat{W}_0^{d_1}] \\
&\qquad T \in \mathcal{T} = \{T \geq 0, \; T\mathbf{1} = \frac{1}{d_2}\mathbf{1}, \; T^\top \mathbf{1} = \frac{1}{d_1}\mathbf{1}\} \; . \quad (10)
\end{aligned}$$

In Eq. 10, we explicitly introduce the optimization process for T when learning W. Thus, when we optimize over the classifier with a fixed semantic map T, we reuse the transformed model as a good prior; when classifier W is fixed, then the optimal transport problem also considers the effect of learning process, i.e., fine tuning the transport plan T w.r.t. the learning performance. In the alternative optimization process, we centralize the bias vector $\mathbf{b}$ as in the previous subsection, then we can get the closed form solution for W as in Eq. 9. When focusing on T, the sub-problem is:

$$\min_{T \in \mathcal{T}} f(T) = \lambda_1 \|W^{d_2} - d_2 T \hat{W}_0^{d_1}\|_F^2 + \lambda_2 \langle T, Q \rangle \; . \quad (11)$$

Different from classical OT problem, Eq. 11 has a squared term over T, which can be regarded as a non-linear regularizer. Therefore, some acceleration techniques, e.g., sinkhorn strategy (Cuturi, 2013), cannot be applied directly. Here we use Bregman Alternating Direction Method of Multipliers (BADMM) (Wang & Banerjee, 2014) to deal with the sub-problem efficiently. Different from ADMM, BADMM replaces the Frobenius norm term in the augmented lagrangian with the bregman divergence, and in a general form, it *linearizes* the loss function to accelerate the optimization process. Introducing an auxiliary variable Z and let $Z = T$, BADMM decomposes the complex constraint domain $\mathcal{T}$ into two parts, i.e., $T \in \mathcal{T}_1 = \{T\mathbf{1} = \frac{1}{d_2}\mathbf{1}, T \geq 0\}$ and $Z \in \mathcal{T}_2 = \{Z^\top \mathbf{1} = \frac{1}{d_1}\mathbf{1}, Z \geq 0\}$. For iteration t, BADMM updates the following three steps.[2]

$$\begin{aligned}
&T^{t+\frac{1}{2}} = (Z^{t\,\frac{\rho}{\rho+\rho_x}} \odot T^{t\,\frac{\rho_x}{\rho+\rho_x}}) \oslash (e^{\frac{U^t + \nabla f(T^t)}{\rho+\rho_x}}) \; , \\
&T^{t+1} = \mathrm{diag}(\frac{1}{d_2 T^{t+\frac{1}{2}} \mathbf{1}}) T^{t+\frac{1}{2}} \; , \; Z^{t+\frac{1}{2}} = T^{t+1} e^{\frac{U^t}{\rho}} \; , \\
&Z^{t+1} = Z^{t+\frac{1}{2}} \mathrm{diag}(\frac{1}{d_1 Z^{t+\frac{1}{2}\top} \mathbf{1}}) \; , \; U^{t+1} = U^t + \rho(T^{t+1} - Z^{t+1}) \; .
\end{aligned}$$

Superscript denotes the iteration of optimization process. $\rho > 0$ and $\rho_x > 0$ are coefficients. U is the dual variable. $\oslash$ denotes the element-wise division. The temporary variable $\nabla f(T^t) = \lambda_1(-2 d_2 W^{d_2} \hat{W}_0^{d_1 \top} + 2 d_2^2 T^t \hat{W}_0^{d_1} \hat{W}_0^{d_1 \top}) + \lambda_2 Q$. Since all updates only involve element-wise calculation, these closed form updates is efficient.

[2] Derivations and convergence analysis are in the supp.

5. Related Work

On the way to the reusable and evolvable properties, researchers investigate from different views. Transfer learning analyzes the knowledge transition from the source to the target domain. Considering the distribution changes between domains, transfer learning focuses on how to extract the source domain information to help the learning process with limited target examples (Pan & Yang, 2010; Si et al., 2010). Heterogeneous transfer learning takes the variations of feature forms between two domains into consideration (Zhu et al., 2011; Aljundi et al., 2015). Structure information or subspaces can be found to link two domains (Shi et al., 2010; Wang & Mahadevan, 2011), where sufficient source domain examples should be provided, even the alignment between instances across domains are required (Kulis et al., 2011). Instead of borrowing knowledge from data, hypothesis transfer aims at using only the source domain homogeneous model to handle the distribution change (Yang et al., 2007; Kuzborskij et al., 2013; Tommasi et al., 2014). Its effectiveness has been proved theoretically in the binary classification case (Kuzborskij & Orabona, 2017). (Hinton et al., 2015; Yang et al., 2015; 2017) transfer the discriminative ability from a related homogeneous strong model to a weak one. Meta-knowledge also facilitates the cross-task transfer, which is usually used in the few-shot learning (Motiian et al., 2017). (Hou & Zhou) first reuses model to deal with the variations on feature space without the alignment assumption, but there needs a specific training strategy on previous tasks. ReForm starts with the theoretical model reuse intuition in the multi-class case, and reuses model from the previous task, even in heterogeneous feature spaces, to improve the performance of the current task with limited examples.

Flexible in incorporating feature meta relationship, Optimal Transport (OT) becomes the main tool in ReForm, which has the ability to align distributions (Villani, 2008; Santambrogio, 2015). With types of solution strategy (Cuturi, 2013; Wang & Banerjee, 2014; Benamou et al., 2015), OT has been successfully applied in various machine learning fields with both its objective measure or the learned transportation plan. For example, in image query (Rubner et al., 1998), document classification (Huang et al., 2016), domain adaptation (Perrot et al., 2016; Courty et al., 2017a;b), and barycenter discovery (Cuturi & Doucet, 2014).

6. Experiments

We first investigate ReForm over synthetic datasets, where feature meta information is generated by Emit to link two tasks together. In addition, reuse performances in different task configurations are studied. Last, we apply ReForm implementations in various real-world applications to show their ability reusing a well-learned model with provided meta information.

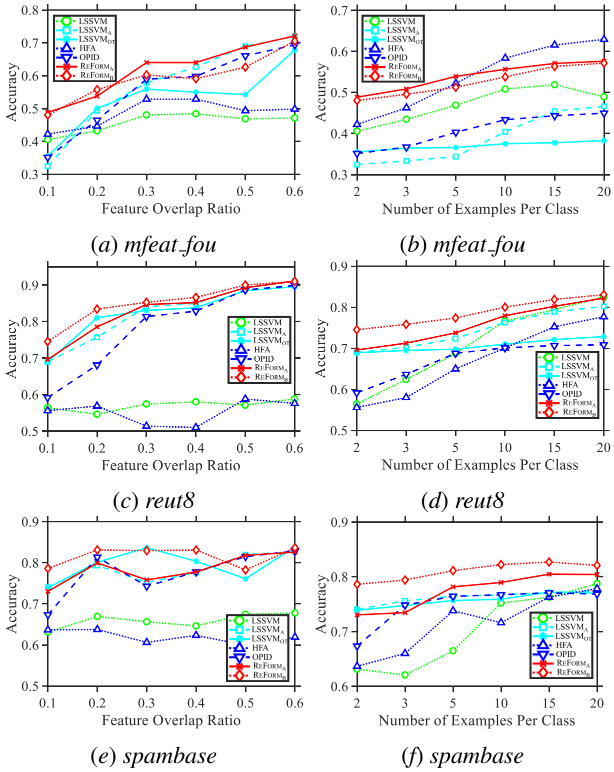
Table 1. Comparisons of classification performance (test accuracy, mean ± std.) including $\text{ReForm}_{\text{A/B}}$. The best performances are in bold. Last two rows list the Win/Tie/Lose counts for ReForm against others with t-test at significance level 95%.

	ReForm_{A}	ReForm_{B}	OPID	LSSVM_{A}	LSSVM_{OT}	SVM
caltech30	**.262±.013**	.248±.011	.128±.042	.256±.009	.219±.006	.123±.017
reut8	.696±.024	**.745±.015**	.592±.183	.690±.015	.689±.015	.570±.024
spambase	.731±.086	**.786±.032**	.673±.196	.741±.032	.739±.037	.644±.126
waveform	**.609±.051**	.497±.036	.516±.077	.514±.022	.459±.041	.344±.024
colic	.619±.074	**.632±.075**	.565±.137	.588±.072	.600±.085	.605±.081
credit-g	.609±.060	.598±.078	**.610±.171**	.606±.059	.558±.098	.545±.130
mfeat_fou	**.488±.035**	.480±.020	.351±.037	.325±.018	.355±.016	.318±.032
optdigits	**.572±.020**	.495±.018	.384±.040	.422±.014	.360±.012	.229±.054
spectf	.569±.128	**.634±.142**	.463±.061	.589±.133	.592±.120	.301±.028
W / T / L	ReForm_{A} vs. others		6 / 3 / 0	5 / 4 / 0	5 / 4 / 0	8 / 1 / 0
W / T / L	ReForm_{B} vs. others		7 / 2 / 0	6 / 1 / 2	8 / 1 / 0	8 / 1 / 0

6.1. General Classification and Parameter Study

We first explore our ReForm approaches on 9 datasets with no meta feature representations. For each dataset, we randomly split features of all examples into three parts, and the dimension proportion of previous task specific features (d_1), current task specific features (d_2), and task shared features (d_3) are 45%, 45%, and 10%, respectively. So there are only 10% percent of overlapping features between former and current tasks. Then half of all examples construct the former task. A linear least square SVM (Ye & Xiong, 2007) classifier is trained on the former task, with parameter tuned by cross-validation. In the remaining half of the current task, only two examples from each class are extracted for training, then 80% of examples are used for test. This process is repeated for 30 trials. The Emit method is conducted in advance to generate feature meta representations using all task-specific instances because of its unsupervised nature.

Our two ReForm implementations, considering adaptive scale and using BADMM solver, are denoted as ReForm_{A} and ReForm_{B}, respectively. We compare our ReForm approaches with various baselines. First, we directly apply linear SVM on the limited current task examples. Adaptive least square SVM (Tommasi et al., 2014) operates as Eq. 1, which requires a prior in the current feature space. Two extensions of homogeneous models can be applied here. After extracting the shared part of the well-trained classifier from the former task, we can pad the remaining part with zero values or with the OT transported prior. Combined these two priors with the adaptive SVM, we get LSSVM_{A} and LSSVM_{OT}. OPID (Hou & Zhou) involves the training in the former task, and ensembles the last stage rectified classifier with stacking. Since with limited target examples, default parameters are used for all methods. This setting also applies to other experiments. Dataset description, comparison results with more methods like HFA (Li et al., 2014), MMDT (Hoffman et al., 2013), OTL (Zhao et al., 2014), and detailed parameter settings can be found in the supp.



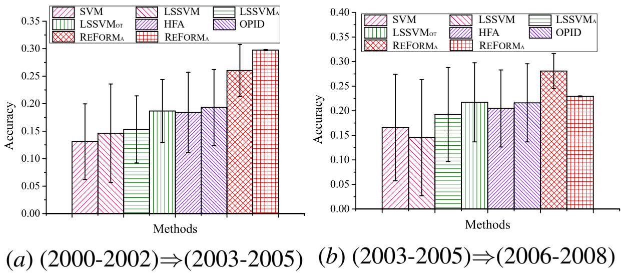
(*a*) *mfeat_fou*   (*b*) *mfeat_fou*

(*c*) *reut8*   (*d*) *reut8*

(*e*) *spambase*   (*f*) *spambase*

Figure 3. Changes of accuracy over *mfeat_fou*, *reut8*, and *spambase*. Plots in the left column show the performance with different feature overlapping ratios (from 0.1 to 0.6); while the right column lists the plots when the amount of training examples increases, i.e., the number of training examples per class from 2 to 20.

Comparison results (test accuracy, mean ± std.) can be found in Table 1. The best performance on each dataset is in bold. We can find that with only a small amount of training examples, SVM cannot perform well. However, after reusing the model from the former task, the performance will improve, which is in accordance with the results in Theorem 1. The adaptive LSSVM with OT-transformed prior sometimes performs better, e.g., in *mfeat_fou* and *spectf*, which shows the OT transformation strategy is able to find good prior between different feature spaces. However, the test accuracy of LSSVM_A could be better sometimes, since the zero prior is sufficient in some cases as in many real problems. OPID uses a stacking strategy to combine previous task co-regularized classifier. Since the number of training examples and overlapping features are limited, OPID cannot perform well. Our ReForm approaches can achieve superior results than other methods in 8/9 datasets, which shows the effectiveness of reusing the heterogeneous model together with limited current task examples to train a good model, and the effectiveness of generating meta information by Emit as well. Since LSSVM_OT equals to ReForm_A without optimizing the scale, the superiority of the latter one validates the necessity of considering the scale. Last



(*a*) (2000-2002)⇒(2003-2005)   (*b*) (2003-2005)⇒(2006-2008)

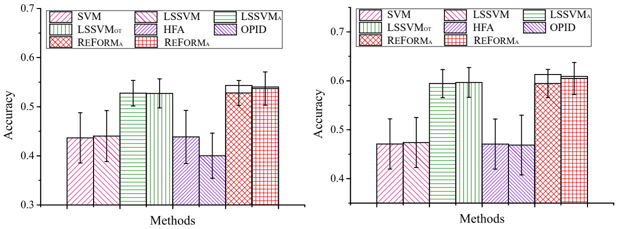
Figure 4. Prediction accuracy and std. for user quality over Amazon Movies and TV review data across different year ranges.

two rows list the Win/Tie/Lose counts for ReForm against other methods with t-test at significance level 95%, which also indicates the effectiveness of our ReForm framework.

We also study the performance of ReForm over tasks with different configurations, i.e., when the amount of shared features between tasks changes and the number of the current task training examples increases. The results are in Fig. 3, where each row corresponds to a dataset. Two plots in one row show the change of feature overlapping ratio from 10% to 60% and instance number per class increases from 2 to 20. The general performance variation reveals an increasing trend in both cases. From Fig. 3, ReForm approaches are in general with the top level performance in different settings, which presents the reusability and evolvability of ReForm in the dynamic environment.

6.2. User Quality Classification

We apply our ReForm implementations to predict whether an Amazon user is high-quality or not given uses' iterations with items. With Amazon user-item click dataset (McAuley et al., 2015; He & McAuley, 2016) over "Movies and TV" sub-category, the user's quality is judged by the helpfulness of his/her review ratings. Average of helpful or not ratios for a user's historical reviews is categorized into 5 levels. Features of users are constructed based on historical behaviors, i.e., review records on items. As the change of time, more items will be added, and out-dated items will be deleted from the online shop. Thus, the user-item interaction features are different in various stages. Time ranges of task 1-3 cover years 2000-2002, 2003-2005, and 2006-2008. About top-1000 popular items in each range are extracted as features. In the current task, only a few labeled users are provided, and the goal is to reuse a well-tuned model from the former task, although with different features, to help the learning of current classifier. For ReForm, online image depiction (CNN extracted features) of a particular item is used as item meta representation. Results are in Fig. 4, which show that ReForm can achieve better performance than other methods. In addition, it is notable that since there are only a few training examples, most compared methods



(*a*) (2013)⇒(2014)

(*b*) (2014)⇒(2015)

Figure 5. Average prediction accuracy and std. on academic paper classification tasks across different year ranges. The blank column at the top of our REFORM implementations show the performance increments after an ensemble step with LSSVM.

possess high variations on results. The prediction accuracy of REFORM$_{\mathrm{B}}$ is stable, which shows its robustness.

6.3. Academic Paper Classification

The "hot-words" in academic papers change with years. For example, new methods would be proposed which are accompanied with new words, and out-dated words vanish. We collect papers from "International Conference on Machine Learning", and then extract TF-IDF features (about 2000-3000 keyword features for each year), one for each word, to do classification tasks. The papers are categorized based on their session names and are organized into 10 classes. Since there are differences between words for papers in different years, this variation of dictionaries leads to examples in each year being with different feature spaces. Word2vec (Mikolov et al., 2013) representation serves as feature meta information. Three subsequent years are investigated, i.e., the model from 2013 corpus helps the learning with papers in 2014, and from 2014 to 2015. Results are listed in Fig. 5. The superior results of REFORM validate its reusability and evolvability with limited examples. Besides, when equipped with an ensemble strategy, i.e., equally averaging the REFORM prediction and the confidence output from LSSVM, REFORM will achieve another performance improvement. The amount of the accuracy increment owing to the ensemble trick is denoted by a blank column on the top of the basic REFORM result in Fig. 5.

6.4. Discussion on Deep Extension

We show the potential usage of our REFORM framework on deep architectures, as illustrated in Fig. 6. Consider the case using multiple fully connected layers where the weights of the first layer are directly compounded with original feature meaning w.r.t. each dimension. When shifting the focus region over images between two tasks, the feature difference hinders the usage of the pre-trained model over the current task. We investigate the 10-class MNIST-Fashion (Xiao et al., 2017) dataset with standard partition.

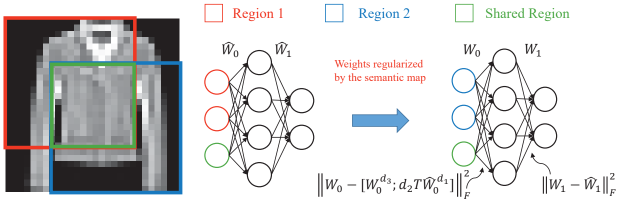

Region 1
Region 2
Shared Region
$\widehat{W}_0$
$\widehat{W}_1$
Weights regularized by the semantic map
W_0
W_1
$\|W_0 - [W_0^{d_3}; d_2 T \widehat{W}_0^{d_1}]\|_F^2$
$\|W_1 - \widehat{W}_1\|_F^2$

Figure 6. Extension of the REFORM idea on neural networks. Layer-wise weights of the current network are regularized by those from a related model. Prior of the first-layer weights corresponding to the changed features can be obtained by REFORM.

For the previous stage, a 4-layer perceptron is trained given the 60000 upper-left 20×20 corner of 28×28 images. In the current task, only bottom-right 20×20 corner images are provided, with only 5 images per class. The model is measured on unused bottom corner images. Although the model achieves a 0.871 accuracy in the previous task, directly applying it on the current task or training over current limited examples degrade the performance a lot, i.e., 0.084 (extreme low since focus on different parts of objects) and 0.564 (since overfitting) respectively. A layer-wise biased regularization strategy like Eq. 1 is used in (Kirkpatrick et al., 2016; Rusu et al., 2016) to overcome the catastrophic forgetting in neural networks. This method, however, only facilitates homogeneous tasks. To construct a suitable prior, we keep coefficients from other layers in the previous model unchanged and transform the first layer coefficients in the model following the REFORM way, where meta features are learned by EMIT. After adding regularizations for each layer biased from the prior, the whole classification accuracy can improve to 0.660 even trained with limited examples. [3]

7. Conclusion

Inspired by the reusable and evolvable properties of *learnware*, we propose the REctiFy via heterOgeneous pRedictor Mapping (REFORM) framework towards robust modeling. First, a well-trained model from the related task is able to be reused to facilitate the current task with the limited amount of training data. In addition, with the Encode Meta InformaTion of features (EMIT) strategy, the generated feature meta information can be leveraged to bridge heterogeneous feature spaces. Thus, the whole framework can adapt models trained with different features sets, which is a practical property handling the dynamic environment. Two implementations of REFORM are investigated on both synthetic and real-world tasks. Experimental results validate their effectiveness, especially with scarce training examples. Future work may include model reuse under more complex environments, e.g., with incremental/decremental classes.

[3] Experimental details and more results can be found in supp.

Acknowledgment

This research was supported by the The National Key R&D Program of China (2018YFB1004300), NSFC (61773198 ,61673201,61751306, 61632004). Authors want to thank reviewers for helpful comments.

References

Aljundi, R., Emonet, R., Muselet, D., and Sebban, M. Landmarks-based kernelized subspace alignment for unsupervised domain adaptation. In *The 28th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 56–63, Boston, MA., 2015.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Benamou, J., Carlier, G., Cuturi, M., Nenna, L., and Peyré, G. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2), 2015.

Bhattarai, B., Sharma, G., and Jurie, F. Cp-mtml: Coupled projection multi-task metric learning for large scale face retrieval. In *The 29th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4226–4235, Las Vegas, NV., 2016.

Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems 30*, pp. 3733–3742. Curran Associates, Inc., 2017a.

Courty, N., Flamary, R., Tuia, D., and Rakotomamonjy, A. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 (9):1853–1865, 2017b.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26*, pp. 2292–2300. Curran Associates, Inc., 2013.

Cuturi, M. and Doucet, A. Fast computation of wasserstein barycenters. In *Proceedings of the 31th International Conference on Machine Learning*, pp. 685–693, Beijing, China, 2014.

Dietterich, T. G. Steps toward robust artificial intelligence. *AI Magazine*, 38(3):3–24, 2017.

Gong, B., Grauman, K., and Sha, F. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 222–230, Atlanta, GA., 2013.

He, R. and McAuley, J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, pp. 507–517, Montreal, Canada, 2016.

Hinton, G. E., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

Hoffman, J., Rodner, E., Donahue, J., Saenko, K., and Darrell, T. Efficient learning of domain-invariant image representations. *CoRR*, abs/1301.3224, 2013.

Hou, C. and Zhou, Z.-H. One-pass learning with incremental and decremental features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. to appear.

Huang, G., Guo, C., Kusner, M. J., Sun, Y., Sha, F., and Weinberger, K. Q. Supervised word mover's distance. In *Advances in Neural Information Processing Systems 29*, pp. 4862–4870. Curran Associates, Inc., 2016.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N. C., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., and Hadsell, R. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016.

Kulis, B., Saenko, K., and Darrell, T. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1785–1792, Colorado Springs, CO., 2011.

Kusner, M. J., Sun, Y., Kolkin, N. I., and Weinberger, K. Q. From word embeddings to document distances. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 957–966, Lille, France, 2015.

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

Kuzborskij, I., Orabona, F., and Caputo, B. From N to N+1: multiclass transfer incremental learning. In *The 26th IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3358–3365, Portland, OR., 2013.

Li, W., Duan, L., Xu, D., and Tsang, I. W. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1134–1148, 2014.

Long, M., Wang, J., Ding, G., Shen, D., and Yang, Q. Transfer learning with graph co-regularization. *IEEE*

Transactions on Knowledge and Data Engineering, 26 (7):1805–1818, 2014.

Maurer, A. A vector-contraction inequality for rademacher complexities. In *Proceedings of the 27th International Conference on Algorithmic Learning Theory*, pp. 3–17, Bari, Italy, 2016.

McAuley, J. J., Targett, C., Shi, Q., and van den Hengel, A. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52, Santiago, Chile, 2015.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

Motiian, S., Jones, Q., Iranmanesh, S. M., and Doretto, G. Few-shot adversarial domain adaptation. In *Advances in Neural Information Processing Systems 30*, pp. 6673–6683. Curran Associates, Inc., 2017.

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

Perrot, M., Courty, N., Flamary, R., and Habrard, A. Mapping estimation for discrete optimal transport. In *Advances in Neural Information Processing Systems 29*, pp. 4197–4205. Curran Associates, Inc., 2016.

Rubner, Y., Tomasi, C., and Guibas, L. J. A metric for distributions with applications to image databases. In *Proceedings of the 6th IEEE International Conference on Computer Vision*, pp. 59–66, Bombay, India, 1998.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

Santambrogio, F. *Optimal transport for applied mathematicians*. Springer, 2015.

Shi, X., Liu, Q., Fan, W., Yu, P. S., and Zhu, R. Transfer learning on heterogenous feature spaces via spectral transformation. In *The 10th IEEE International Conference on Data Mining*, pp. 1049–1054, Sydney, Australia, 2010.

Si, S., Tao, D., and Geng, B. Bregman divergence-based regularization for transfer subspace learning. *IEEE Transaction on Knowledge and Data Engineering*, 22(7):929–942, 2010.

Sugiyama, M. and Kawanabe, M. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

Tommasi, T., Orabona, F., and Caputo, B. Learning categories from few examples with multi model knowledge transfer. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 36(5):928–941, 2014.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Wang, C. and Mahadevan, S. Heterogeneous domain adaptation using manifold alignment. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1541–1546, Barcelona, Catalonia, 2011.

Wang, H. and Banerjee, A. Bregman alternating direction method of multipliers. In *Advances in Neural Information Processing Systems 27*, pp. 2816–2824. Cambridge, MA.: MIT Press, 2014.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Yang, J., Yan, R., and Hauptmann, A. G. Cross-domain video concept detection using adaptive svms. In *Proceedings of the 15th International Conference on Multimedia*, pp. 188–197, Augsburg, Germany, 2007.

Yang, Y., Ye, H.-J., Zhan, D.-C., and Jiang, Y. Auxiliary information regularized machine for multiple modality feature learning. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 1033–1039, Buenos Aires, Argentina, 2015.

Yang, Y., Zhan, D.-C., Fan, Y., Jiang, Y., and Zhou, Z.-H. Deep learning for fixed model reuse. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pp. 2831–2837, San Francisco, CA., 2017.

Ye, J. and Xiong, T. SVM versus least squares SVM. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pp. 644–651, San Juan, Puerto Rico, 2007.

Zhao, P., Hoi, S. C., Wang, J., and Li, B. Online transfer learning. *Artificial Intelligence*, 216:76–102, 2014.

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

Zhu, Y., Chen, Y., Lu, Z., Pan, S. J., Xue, G.-R., Yu, Y., and Yang, Q. Heterogeneous transfer learning for image classification. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pp. 1304–1309, San Francisco, CA., 2011.