Supplemental Material for Heterogeneous Few-Shot Model Rectification with Semantic Mapping

Han-Jia Ye, De-Chuan Zhan, Yuan Jiang, and Zhi-Hua Zhou, *Fellow, IEEE*

Abstract

This is the supplemental material for the "Heterogeneous Few-Shot Model Rectification with Semantic Mapping". In the manuscript, we propose the REctiFy via heterOgeneous pRedictor Mapping (REFORM) framework to reuse a related model with heterogeneous features or classes, so that the current task model could be efficiently constructed by few-shot training data. There are four parts in this supplementary. First, we provide concrete proofs for the theoretical results in the paper (cf. Section 1). Second, we derive the optimization steps for the Bregman ADMM solver of our approach (cf. Section 2). Then, we discuss how to obtain the meta-representation, i.e., encoding the meta-information of features and classes (cf. Section 3). Last, we list the statistics of the datasets and additional experimental results (cf. Section 4).

1 Theoretical Analysis of the Homogeneous Model Reuse

Recall the problem setting, we consider a C-class dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ sampled from the latent distribution $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ correspond to the instance and label distribution, respectively. In each pair, instance $\mathbf{x}_i \in \mathbb{R}^d$, and label vector $\mathbf{y}_i \in \{-1, 1\}^C$. The position of the value 1 in $\mathbf{y}_i$ denotes the class label. In this theoretical analysis, we assume all the instances are in the homogeneous form without loss of generality. In other words, there is a constant value 1 at the end of each instance representation (so we do not need to consider the bias in the classifier). In addition, we can bound each instance by $\|\mathbf{x}_i\| \leq \chi$ for $i = 1, \ldots, N$. χ is a positive scalar.

Consider the linear classifier $W \in \mathbb{R}^{d \times C}$ maps an instance to a label, whose columns correspond to each class. Given training data $\mathcal{D}$, the classifier can be learned by the following form:

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

where $f(\mathbf{x}_i) = W^\top \mathbf{x}_i \in \mathbb{R}^C$ is the model prediction over the instance $\mathbf{x}_i$. Loss function $\ell(\cdot) : \mathbb{R}^C \to \mathbb{R}_+$ measures the difference between *vector form* prediction and the true label[1]. It is notable that in the vector form loss function, prediction confidence for multiple classes can be considered together, and it is different from the binary case. Instead of learning the linear predictor directly, we focus on the helpfulness of *reusing* a related model W_0 from the same feature space:

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

$\epsilon_N(W)$ is the empirical risk of the learning problem, which is depend on the totally N examples. The expected risk of Eq. 1 can be formulated as:

$$\epsilon(W) = \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim \mathcal{Z}}[\ell(f(\mathbf{x}) - \mathbf{y})] \;. \tag{2}$$

Generalization analysis focus on both the gap and the rate between $\epsilon_N(W)$ and $\epsilon(W)$ [1], [2]. We prove that the consideration of a well-trained related model W_0 improves the learning efficiency, i.e., model reuse improves the convergence rate from $\epsilon_N(W)$ to $\epsilon(W)$.

- *H.-J. Ye, D.-C. Zhan, Y. Jiang and Z.-H. Zhou are with National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, 210023, China.*
  E-mail: {yehj,zhandc,jiangy,zhouzh}@lamda.nju.edu.cn

[1] We use $\mathbb{R}_+$ to denote the non-negative domain of real numbers.

Theorem 1. Consider a C-class learning problem from $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ as in Eq. 1. Given a L-Lipschitz vector valued loss function w.r.t. Euclidean norm which can be bounded by M, then for every model $W \in \mathcal{W}$ and $0 < \delta < 1$, with probability at least $1 - \delta$, we have:

$$\epsilon(W) \leq \epsilon_N(W) + \frac{C_1}{N} + C_2\sqrt{\frac{\epsilon(W_0)}{N}} \ , \tag{3}$$

where $C_1 = \frac{2M \log 1/\delta}{3} + 4MLC\chi \log 1/\delta$, and $C_2 = \frac{4LC\chi+2}{\sqrt{\lambda}} + \sqrt{2M \log 1/\delta}$.

The main steps of the proof are non-trivial extensions of the previous ones [1], [2]. There are three steps in the proof. First, we prove that the generalization gap can be bounded by the Rademacher Complexity of the model given a general ceiling for the expected risk of the old model; then we determine the complexity of the model and its relationship with the number in the task; finally, we explain that how the learning problem in Eq. 1 satisfy previous assumptions.

Proof: We first introduce the following lemma as the main tool:

Lemma 1. [3] Assume the $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{y}_i)$ are i.i.d. according to $\mathcal{Z}$. Let $\mathcal{F}$ be a countable set of functions from $\mathcal{Z}$ to $\mathbb{R}$, and assume that all functions f are measurable, square-integrable, satisfying $\mathbb{E}[f] = 0$. If $\sup_{f \in \mathcal{F}} f \leq 1$, then denote $Z = \sup_{f \in \mathcal{F}} \sum_{i=1}^N f(\mathbf{z}_i)$. If $\sigma > 0$ and $\sigma^2 \geq \sup_{f \in \mathcal{F}} \mathrm{Var}[f(\mathbf{z}_i)]$, let $v = N\sigma^2 + 2\mathbb{E}[Z]$. Then $\forall x > 0$, we have $\Pr(Z \geq \mathbb{E}[Z] + \sqrt{2xv} + \frac{x}{3}) \leq e^{-x}$.

For a bounded loss function $0 \leq \ell(f(\mathbf{x}) - \mathbf{y}) \leq M$, we consider the property of the empirical process $Z = \frac{N}{2M} \sup_{W \in \mathcal{W}} \epsilon(W) - \epsilon_N(W)$. With the help of Lemma 1, we have with probability at least $1 - \delta$,

$$Z \leq \mathbb{E}[Z] + \sqrt{2v \log 1/\delta} + \frac{\log 1/\delta}{3} \ , \tag{4}$$

where $\sigma^2 \geq \sup_{f \in \mathcal{F}} \mathrm{Var}[\frac{1}{2M}(\ell(f(\mathbf{x}), \mathbf{y}) - \mathbb{E}[\ell(f(\mathbf{x}), \mathbf{y})])]$, and $v = N\sigma^2 + 2\mathbb{E}[Z]$. Now the problem remains to determine the value of σ and upper bound $\mathbb{E}[Z]$.

σ^2 is the upper bound of variance. Together with the bound of loss function, it can be determined by $\frac{1}{4M} \sup_{f \in \mathcal{F}} \mathbb{E}[\ell(f(\mathbf{x}), \mathbf{y})] = \frac{1}{4M} \sup_{W \in \mathcal{W}} \mathbb{E}[\epsilon_N(W)] \leq \frac{1}{4M} \mathbb{E}[\sup_{W \in \mathcal{W}} \epsilon_N(W)] \leq \frac{1}{4M} \mathbb{E}[\epsilon_N(W_0)] = \frac{1}{4M} \epsilon(W_0)$. Here we utilize the fact that the optimal value of the empirical loss function will be lower than the value at W_0.

To bound $\mathbb{E}[Z]$, we introduce Rademacher random variable σ_i drawing independently from {-1,1} with equal probability. Using symmetrization techniques [4], we have

$$\begin{aligned}
\mathbb{E}[Z] &= \frac{N}{2M} \mathbb{E}[\sup_{W \in \mathcal{W}} \epsilon(W) - \epsilon_N(W)] \\
&= \frac{N}{2M} \mathbb{E}[\sup_{W \in \mathcal{W}} \mathbb{E}[\epsilon_N(W)] - \epsilon_N(W)] \\
&\leq \frac{1}{M} \mathbb{E}_{\mathbf{x}_i, \mathbf{y}_i, \sigma_i}[\sup_{W \in \mathcal{W}} \sum_{i=1}^N \sigma_i \ell(f(\mathbf{x}_i) - y_i)] \ .
\end{aligned}$$

Since $\ell(\cdot)$ is a vector-valued Lipschitz continuous function, we can use the following contraction lemma to get rid of the loss in above inequality:

Lemma 2. [5] Let $\mathcal{X}$ be any set, $(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ are drawn from $\mathcal{X}$. Let members of $\mathcal{F}$ take values in $\mathbb{R}^C$ and with component functions $f_c(\cdot)$. If ℓ is L-Lipschitz from $\mathbb{R}^C$ with Euclidean norm to $\mathbb{R}$, and σ_{ic} are an $N \times C$ matrix of independent Rademacher variables, we have the vector contraction inequality

$$\mathbb{E} \sup_{f \in \mathcal{F}} \sum_{i=1}^N \sigma_i \ell(f(\mathbf{x}_i)) \leq \sqrt{2} L \mathbb{E} \sup_{f \in \mathcal{F}} \sum_{i=1}^N \sum_{c=1}^C \sigma_{ic} f_c(\mathbf{x}_i) \ . \tag{5}$$

Lemma 2 helps remove the loss function, and decomposes the sum over Rademacher variables into multiple class specific sums. Then we can transform $\mathbb{E}[Z]$ by:

$$\begin{aligned}
\mathbb{E}[Z] \leq\ & \frac{\sqrt{2}L}{M}\mathbb{E}[\sup_{W\in\mathcal{W}}\sum_{i=1}^{N}\sum_{c=1}^{C}\sigma_{ic}f_c(\mathbf{x}_i)] \\
=\ & \frac{\sqrt{2}L}{M}\mathbb{E}[\sup_{W\in\mathcal{W}}\sum_{i=1}^{N}\sum_{c=1}^{C}\sigma_{ic}\mathbf{w}_c^\top\mathbf{x}_i] \\
=\ & \frac{\sqrt{2}L}{M}\mathbb{E}[\sup_{W\in\mathcal{W}}\sum_{c=1}^{C}\langle\mathbf{w}_c,\sum_{i=1}^{N}\sigma_{ic}\mathbf{x}_i\rangle] \qquad (6) \\
=\ & \frac{\sqrt{2}L}{M}\mathbb{E}[\sup_{W\in\mathcal{W}}\sum_{c=1}^{C}\langle\mathbf{w}_c-\mathbf{w}_{0,c},\sum_{i=1}^{N}\sigma_{ic}\mathbf{x}_i\rangle] \qquad (7) \\
\leq\ & \frac{\sqrt{2}L}{M}\mathbb{E}[\sup_{W\in\mathcal{W}}\|W-W_0\|_F\sum_{c=1}^{C}\|\sum_{i=1}^{N}\sigma_{ic}\mathbf{x}_i\|] \\
\leq\ & \frac{\sqrt{2}LC}{M}\mathbb{E}[\sqrt{\frac{\epsilon_N(W_0)}{\lambda}}]\sqrt{N}\chi \\
\leq\ & \frac{\sqrt{2}LC\sqrt{N}\chi}{M}\sqrt{\frac{\epsilon(W_0)}{\lambda}}\ .
\end{aligned}$$

In above derivations, Eq. 6 results from the linearity of inner product. In Eq. 7, we introduce the *constant* vector $\mathbf{w}_{0,c}$, which is the c-th column of prior matrix W_0. It is notable that since this is a constant term, when combined with Rademacher variables it only outputs zero.

Therefore,

$$\begin{aligned}
\frac{2M}{N}\sqrt{2v} &\leq \sqrt{\frac{8M^2}{N}\sigma^2+\frac{16\sqrt{2}MLC\chi}{N}\sqrt{\frac{\epsilon(W_0)}{\lambda N}}} \\
&\leq \sqrt{\frac{2M\epsilon(W_0)}{N}}+\frac{4MLC\chi}{N}+2\sqrt{\frac{\epsilon(W_0)}{\lambda N}}\ ,
\end{aligned}$$

where we use the inequality for $a,b>0$, $\sqrt{a+b}\leq\sqrt{a}+\sqrt{b}$. Combining previous results together, we have:

$$\begin{aligned}
\sup_{W\in\mathcal{W}}\epsilon(W)-\epsilon_N(W) \leq\ & \frac{4LC\chi}{\sqrt{N}}\sqrt{\frac{\epsilon(W_0)}{\lambda}}+\frac{2M\log 1/\delta}{3N} \\
&+\sqrt{\frac{2M\epsilon(W_0)\log 1/\delta}{N}}+\frac{4MLC\chi\log 1/\delta}{N}+2\sqrt{\frac{\epsilon(W_0)}{\lambda N}}\ .
\end{aligned}$$

Define $C_1=\frac{2M\log 1/\delta}{3}+4MLC\chi\log 1/\delta$, and $C_2=\frac{4LC\chi+2}{\sqrt{\lambda}}+\sqrt{2M\log 1/\delta}$, we have

$$\sup_{W\in\mathcal{W}}\epsilon(W)-\epsilon_N(W)\leq\frac{C_1}{N}+C_2\sqrt{\frac{\epsilon(W_0)}{N}}\ . \qquad \blacksquare$$

We can first bound the domain of the linear classifier W. Denote the empirical optimal solution of Eq. 1 as W^*. Due to the optimality of the empirical objective at W^*, we have

$$\begin{aligned}
&\epsilon_N(W^*)+\lambda\|W^*-W_0\|_F^2 \\
&\qquad\leq\epsilon_N(W_0)+\lambda\|W_0-W_0\|_F^2=\epsilon_N(W_0)\ , \qquad (8)
\end{aligned}$$

thus $\|W^*-W_0\|_F\leq\sqrt{\frac{\epsilon_N(W_0)}{\lambda}}$. From Eq. 8, there is a loss constraint for the optimal solution W^*, i.e., $\epsilon_N(W^*)\leq\epsilon_N(W_0)$. In addition, we also have

$$\begin{aligned}
&\|W^*\|_F=\|W^*-W_0+W_0\|_F \qquad (9) \\
&\qquad\leq\|W^*-W_0\|_F+\|W_0\|_F\leq\sqrt{\frac{\epsilon_N(W_0)}{\lambda}}+\|W_0\|_F\ .
\end{aligned}$$

So the optimal solution is in a bounded domain $W^*\in\mathcal{W}=\{W\in\mathbb{R}^{d\times C},\|W-W_0\|_F\leq\sqrt{\frac{\epsilon_N(W_0)}{\lambda}}\ ,\epsilon_N(W)\leq\epsilon_N(W_0)\}$.

From the results, it can be found there is in general a $\mathcal{O}(\frac{1}{\sqrt{N}})$ convergence rate for the learning problem in Eq. 1. Thus, when the number of examples is large enough, the empirical risk approaches the expected risk, and learning by empirical risk actually optimizes over the true distribution. This convergence rate also indicates how large sample size is required for a particular learning precision. In other words, given a fixed number of training samples, the faster the convergence rate, the better the performance of a model can obtain with these limited examples.

It can also be found that when the well-trained former task model W_0 acts as a good prior, then its expected risk will approach zero, which *gets rid of* the $\mathcal{O}(\frac{1}{\sqrt{N}})$ term on the r.h.s. of the Eq. 3. It indeed improve the convergence rate of the learning problem from the order $\mathcal{O}(\frac{1}{\sqrt{N}})$ to the order $\mathcal{O}(\frac{1}{N})$. Thus, homogeneous model reuse, in this case, can get lower sample complexity.

Remark 1. We assume two tasks have the same feature space (homogeneous feature space) and the label set (for the same C-class classification task). The distribution change between instances across two tasks could be measured by the $\epsilon(W_0)$, i.e., how the provided model prior W_0 fits the current task. Therefore, if two tasks are related, then a well-trained model from the former task has a low value for $\epsilon(W_0)$, which will help a lot. It explains why a *"related"* and *"well-trained"* model facilitates the current learning process. "Well-trained" means the old task model works well on its original distribution, which has a low expected error; while the "relatedness" guarantees that the low expected error on the old distribution can be persisted on the distribution of the current task, so that we have $\epsilon(W_0)$ approaching zero.

Remark 2. The assumption of the theorem is weak. For least square loss $\ell(\cdot) = \|\cdot - \mathbf{y}\|_F^2$, it is only L-Lipschitz in a constrained domain but not in general [6]. But from the bound of norm over W together with the bound χ over instances, we can find that the predictions over all instances can be bounded.

Remark 3. Hypothesis transfer theorem [2] analyzes the similar phenomenon given a well-trained model from the related homogeneous task, i.e., reusing a good model can improve the order of the convergence rate. However, our analysis focuses on a more general *multi-class* case, which is really common in real-world tasks. Our analysis is a non-trivial extension w.r.t. the binary case in [2]. In addition, in our proof, there does not need smooth assumption of the loss function.

Remark 4. The condition of new hypothesis $W^* \in \mathcal{W} = \{W \in \mathbb{R}^{d\times C}, \|W - W_0\|_F \leq \sqrt{\frac{\epsilon_N(W_0)}{\lambda}}, \epsilon_N(W) \leq \epsilon_N(W_0)\}$ indicates that the optimal new hypothesis is constrained in the *neighborhood* of the prior W_0. It is reasonable since the provided model can be reused if it is close to the optimal solution of the new task. The larger the $\epsilon_N(W_0)$, the wider the neighbor area, but the criterion $\epsilon(W_0)$ may also be large.

2 Derivation of BADMM Solver

This section presents the detailed derivation of the Bregman ADMM solver for ReForm implementation. With Q as the cost matrix, the optimization problem is:

$$\begin{aligned} \min_{W,\mathbf{b},T} \; & \|Y - XW - \mathbf{1}\mathbf{b}^\top\|_F^2 + \lambda_1\|W - W_0\|_F^2 + \lambda_2\langle T, Q\rangle \\ s.t. \;\; & W_0 = [\hat{W}_0^{\bar{d}}; dT\hat{W}_0^{d'}] \\ & T \in \mathcal{T} = \{T \geq 0,\; T\mathbf{1} = \frac{1}{d}\mathbf{1},\; T^\top\mathbf{1} = \frac{1}{d'}\mathbf{1}\}\;. \end{aligned} \tag{10}$$

We solve this problem in an alternative manner. By introducing a centralization matrix $H = I - \frac{1}{N}\mathbf{1}\mathbf{1}^\top$, the subproblem over variable W can be reformulated as:

$$\min_W \|HY - HXW\|_F^2 + \lambda_1\|W - W_0\|_F^2 \;, \tag{11}$$

which can be solved with the closed form:

$$W = (X^\top HX + \lambda_1 I)^{-1}(\lambda_1 W_0 + X^\top HY) \;. \tag{12}$$

When the dimension of features is large enough, the closed form can be simplified by Woodbury Identity. The subproblem in Eq. 11 can also be handled with gradient descent efficiently.

When we optimize over T, the sub-problem can be reformulated as:

$$\min_{T\in\mathcal{T}} f(T) = \lambda_1\|W^d - dT\hat{W}_0^{d'}\|_F^2 + \lambda_2\langle T, Q\rangle \;. \tag{13}$$

W^d is the task specific feature part of current solution W, i.e., the last d rows of the learned W, which is fixed in this stage. Different from the classical OT problem, Eq. 13 has a squared term over T (the first term in Eq. 13), which can be regarded as a non-linear regularizer. Therefore, some acceleration techniques, e.g., sinkhorn strategy [7], cannot be applied directly. Here we use Bregman Alternating Direction Method of Multipliers (BADMM) [8] to deal with the sub-problem efficiently.

Introducing an auxiliary variable Z and add the constraint $Z = T$, BADMM decomposes the complex constraint domain $\mathcal{T}$ into two parts:

$$\begin{aligned} &\min_{T,Z} f(T) = \lambda_1 \| W^d - dT\hat{W}_0^{d'} \|_F^2 + \lambda_2 \langle T, Q \rangle \\ &s.t. \;\; T - Z = 0 \; , \\ &\qquad T \in \mathcal{T}_1 = \{ T\mathbf{1} = \frac{1}{d}\mathbf{1}, T \geq 0 \} \; , \\ &\qquad Z \in \mathcal{T}_2 = \{ Z^\top \mathbf{1} = \frac{1}{d'}\mathbf{1}, Z \geq 0 \} \; . \end{aligned} \tag{14}$$

Based on Eq. 14, ADMM can progress by solving from dual with augmented lagrangian. The key difference between BADMM and ADMM is that BADMM replaces the Frobenius norm term in the traditional augmented lagrangian by the Bregman divergence.

$$L_\rho^\phi(T, Z, U) = f(T) + \langle U, T - Z \rangle + \rho B_\phi(T, Z) \; . \tag{15}$$

Here U is the dual variable, which is the same size as T and Z. ρ is a non-negative parameter. $B_\phi(T, Z)$ is the Bregman divergence between two variables T and Z w.r.t a strictly convex differential function ϕ, which is defined as:

$$B_\phi(T, Z) = \phi(T) - \phi(Z) - \langle \nabla\phi(Z), T - Z \rangle \; .$$

$B_\phi(T, Z)$ reveals the divergence between T and Z, which is non-negative, and convex w.r.t. its first argument. BADMM solves Eq. 15 in an alternative manner, i.e., in the t-th step, it iterates and updates following three steps:

$$\begin{aligned} T^{t+1} &= \mathop{\arg\min}_{T \in \mathcal{T}_1} f(T) + \langle U_t, T - Z^t \rangle + \rho B_\phi(-T, -Z^t) + \rho_T B_{\phi_T}(T, T^t) \; , \\ Z^{t+1} &= \mathop{\arg\min}_{Z \in \mathcal{T}_2} \langle U^t, T^{t+1} - Z \rangle + \rho B_\phi(-Z, -T^{t+1}) \; , \\ U^{t+1} &= U^t + \rho(T^{t+1} - Z^{t+1}) \; . \end{aligned}$$

As in general ADMM, the following updates of Z and U utilize the previous update variable. Considering the convexity of Bregman divergence, current optimization variable is placed in the first term of Bregman regularizer. In the update of T, BADMM introduces another Bregman term w.r.t. ϕ_T, which can be used to *linearize* the loss function to accelerate the optimization process [8]. ρ_T is the corresponding parameter, we set it the same as ρ in our implementation. Assume there exists $\phi_T(T) = \psi(T) - \frac{1}{\rho_T} f(T)$, the updates over T can be transformed to:

$$T^{t+1} = \mathop{\arg\min}_{T \in \mathcal{T}_1} \langle \nabla f(T^t), T - T^t \rangle + \langle U_t, T \rangle + \rho B_\rho(-T, -Z^t) + \rho_x B_\psi(T, T^t) \; ,$$

Since $B_{\phi_T}(\cdot, \cdot)$ captures the high order information of $f(T)$, the optimization can be reduced to a linear form. By implementing Bregman term with KL-divergence, i.e., $\mathrm{KL}(Z, T) = \sum Z \odot \log Z \oslash T$, where $\odot$ and $\oslash$ denote the element-wise product and division respectively, the updates can be simplified as follows:

$$\begin{aligned} T^{t+1} &= \mathop{\arg\min}_{T \in \mathcal{T}_1} \langle \nabla f(T^t), T \rangle + \langle U_t, T \rangle + \rho \mathrm{KL}(T, Z^t) + \rho_x \mathrm{KL}(T, T^t) \; , \\ Z^{t+1} &= \mathop{\arg\min}_{Z \in \mathcal{T}_2} -\langle U^t, Z \rangle + \rho \mathrm{KL}(Z, T^{t+1}) \; , \\ U^{t+1} &= U^t + \rho(T^{t+1} - Z^{t+1}) \; . \end{aligned}$$

The temporary gradient can be computed as

$$\nabla f(T^t) = \lambda_1(-2dW^d \hat{W}_0^{d'\top} + 2d^2 T^t \hat{W}_0^{d'} \hat{W}_0^{d'\top}) + \lambda_2 Q \; .$$

Benefited from the constraints decomposition and $\mathrm{KL}(\cdot, \cdot)$ regularizer, these updates have closed form solution [9].

$$\begin{aligned} T^{t+\frac{1}{2}} &= (Z^{t\,\frac{\rho}{\rho+\rho_x}} \odot T^{t\,\frac{\rho_x}{\rho+\rho_x}}) \oslash (e^{\frac{U^t + \nabla f(T^t)}{\rho+\rho_x}}) \; , \\ T^{t+1} &= \mathrm{diag}(\frac{1}{dT^{t+\frac{1}{2}}\mathbf{1}}) T^{t+\frac{1}{2}} \; , \\ Z^{t+\frac{1}{2}} &= T^{t+1} e^{\frac{U^t}{\rho}} \; , \; Z^{t+1} = Z^{t+\frac{1}{2}} \mathrm{diag}(\frac{1}{d' Z^{t+\frac{1}{2}\top}\mathbf{1}}) \; . \end{aligned}$$

With these updates, we ensure that the optimization variables satisfy the constraints. Since all updates only involve element-wise calculation, the whole process is efficient.

The convergence guarantee of this BADMM process can be found in [8]. Since in each sub-problem the objective function will decrease, the whole optimization process will converge at last.

Remark 5. The KL divergence projection step [9] in the BADMM updates actually imposes a smooth assumption over the solutions like [7]. It needs the input variable be positive. So the initial solution must be with only positive values. We can

	Name	N_f	N	d'	$\bar{d}$	d	C
Synthetic	caltech30	2739	2739	225	50	225	30
	colic	184	184	27	6	27	2
	credit-g	500	500	27	7	27	2
	mfeat_fou	1000	1000	34	8	34	10
	optdigits	2810	2810	28	8	28	10
	reut8	3835	3835	225	50	225	8
	spambase	2301	2300	25	7	25	2
	spectf	175	174	19	6	19	2
	waveform	2500	2500	18	4	18	3
Amazon	Amazon1997	1423	13273	505	441	557	5
	Amazon2000	13273	16016	665	333	673	5
	Amazon2003	16016	18139	741	265	734	5
	Amazon2006	18139	19906	744	255	740	5
	Amazon2009	19906	57912	688	307	692	5
Corpus	ICML2013	202	231	261	2613	624	10
	ICML2014	231	194	548	2689	325	10
	ICML2015	194	276	165	2849	1049	10

Table 1: The basic information of some datasets when reusing a model with Heterogeneous Features (HF).

add a small positive perturbation over the solution at first. The behavior as in [7] also makes the whole optimization process fast.

Remark 6. Since there is a non-linear term in the objective function, original OT solver cannot be applied in this case. To take advantage of existing OT solver, linearize the objective is a practical way. Frank-Wolfe [10], a.k.a. conditional gradient descent, can also be used to solve this subproblem [11]. The BADMM solution can achieve faster convergence rate during optimization [8].

Remark 7. Here we briefly analyze the computational complexity of two ReForm implementations. Since there are closed form solutions in all steps of ReForm$_{A/B}$ solver, which ensures the efficiency of the optimization process. For ReForm$_A$, each step has the same computational complexity as in ridge regression. Besides, when solving class-specific weights, the vector form solution can be computed in parallel; while in ReForm$_B$, the number of outer iterations follows the convergence rate of BADMM [8], and for inner updates, since all updates formulations are elementwise, the complexity is proportional to the input size.

3 Discussion

There are two types of heterogeneous between the old and the current tasks, i.e., the related task with Heterogeneous Features (**HF**) and Heterogeneous Classes (**HC**). One main step in ReForm. To bridge the two types of heterogeneity between tasks, we propose to Encode the Meta InformaTion (Emit) for either features or classes.

Emit for the HF case: We assume there is no rapid change of features between tasks, and take advantage of the shared features (with dimensionality $\bar{d}$) as the dictionary to encode the meta-representation of task-specific features. In real applications, the shared features are usually constructed by those features with general meaning. Like in document classification, feature sets correspond to words and intersected features maybe those high-frequency words in documents; while for different branches of a company aiming at the same task, the common features are location agnostic and usually extracted based on some general protocols. In the scenarios, there is no interaction between two feature spaces, Emit for HF can turn to some publicly obtained intermediaries to bridge the heterogeneous feature space gap.

Some previous approaches handle the knowledge transfer between heterogeneous spaces based on the raw data from the former task [12], [13], [14]. For example, a mapping between feature spaces is learned via pairs of heterogeneous examples. In ReForm, only the cost matrix relied on meta-features is used in the current task training to obtain a semantic map. Since it is hard to recover original raw features from this cost matrix, raw data privacy is preserved when reusing the heterogeneous model with the transition cost matrix [15].

The meta representations of features in Emit provide a way to depict the change of the environment and constitute one fundamental way for the evolvability of a model. First, the meta-features enable the linkages of different sets of features and even apply the linkage to the model space. Besides, the variation on features can be revealed by the distribution variation of meta representations, so the model can perceive the change of environment by detecting the change in the meta-space.

Emit for the HC case: In this case, Emit uses the prototype (a.k.a. center) of each class as the class-level meta-information. Since all the instances are embedded in the same feature space, the similarity between class centers reveals the similarity between their corresponding classifiers. Although there are only a limited number of instances in the current task, averaging the same-class instance embedding still achieve accurate estimation, which has been validated in [16], [17] and our few-shot classification experiments.

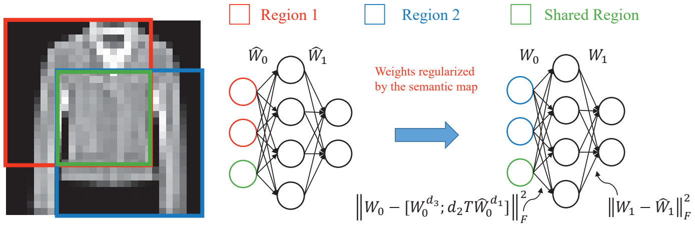
□ Region 1
□ Region 2
□ Shared Region
$\widehat{W}_0$
$\widehat{W}_1$
Weights regularized by the semantic map
W_0
W_1
$\left\|W_0 - [W_0^{d_3}; d_2 T \widehat{W}_0^{d_1}]\right\|_F^2$
$\left\|W_1 - \widehat{W}_1\right\|_F^2$

Figure 1: Extension of the ReForm idea on neural networks. Layer-wise weights of the current network are biased regularized. Prior of the first-layer weights can be obtained by ReForm.

4 Additional Experiments

4.1 Dataset Information

The basic information of some datasets when reusing a model with Heterogeneous Features (HF) is listed in Table. 1. There are three groups of datasets, namely, the synthetic datasets with no meta information, Amazon users quality classification datasets using item image as the meta information, and International Conference on Machine Learning (ICML) academic paper corpus using word2vec [18] as the feature meta information. In the table, N_f and N are the number of examples in former and current tasks. C is the number of class in both tasks. d' and d are the dimension for former and current task specific features, while $\bar{d}$ is the shared feature dimensionality between two tasks.

4.2 Text Classification Tasks

We provide additional experimental results applying ReForm on deep text classification tasks upon the Reuters dataset. There are 45 classes in total in the former and the current tasks. Since the most frequent words could change through time, the input features vary accordingly.

For the first stage, we set the maximum size of the word dictionary as 800, and skip the most 50 frequent words. Multi-layer perceptron with one hidden layer of size 512 and ReLU activation is used as the basic architecture. After the hidden layer, we add dropout layers with rate 0.2. The loss is computed by cross-entropy. The weights of the model are initialized by the Xavier strategy if without further mentions. A well-trained model is tuned on the first 8982 instances with the Adam optimizer [19], batch size equals 32, and the number of epochs is 15, which can achieve 0.801 accuracy. For the second stage, we use the 2nd part of the whole dataset, set the maximum size of the word dictionary as 800, and skip the most 300 frequent words. Thus there are 500 shared features between these two tasks. We try a one-shot learning task, i.e., there is only one instance sampled from each class in the current task. This process is repeated five times, and the average test accuracy on the remaining 2246 instances is reported.

Extracting the shared feature part of the 1st layer weights from the former task model and padding remaining part by zero, this direct strategy cannot work and only gets a 0.009 accuracy. After constructing the feature cost matrix by word2vec pairwise distance matrix as in the academic document classification task, we can transform the previous specific first layer weights to the current task. This transformation strategy can achieve 0.254 accuracy even without training. Using this OT transformed weights to warm-start the MLP training on the current task can achieve 0.559 accuracy, while warm start training with the zero-padding weights gets 0.547. Using a biased OT transformed regularizer in Fig. 1, the new task training can be trained based on transformed previous well-trained value and can achieve 0.568 accuracy. Since the training of MLP is a non-convex difficult task, the improvement of deep attention of ReForm is not as obvious as the previous linear classifier adaptation. But the results still validate that reuse a heterogeneous model in the ReForm way could be an effective approach when facing scarce training examples.

References

[1] M. Perrot and A. Habrard, “A theoretical analysis of metric hypothesis transfer learning,” in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015, pp. 1708–1717.
[2] I. Kuzborskij and F. Orabona, “Fast rates by transferring from auxiliary hypotheses,” *Machine Learning*, vol. 106, no. 2, pp. 171–195, 2017.
[3] O. Bousquet, “A bennett concentration inequality and its application to suprema of empirical processes,” *Comptes Rendus Mathematique*, vol. 334, no. 6, pp. 495–500, 2002.
[4] P. L. Bartlett and S. Mendelson, “Rademacher and gaussian complexities: Risk bounds and structural results,” *Journal of Machine Learning Research*, vol. 3, pp. 463–482, 2002.
[5] A. Maurer, “A vector-contraction inequality for rademacher complexities,” in *Proceedings of the 27th International Conference on Algorithmic Learning Theory*, Bari, Italy, 2016, pp. 3–17.
[6] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
[7] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transport,” in *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013, pp. 2292–2300.
[8] H. Wang and A. Banerjee, “Bregman alternating direction method of multipliers,” in *Advances in Neural Information Processing Systems 27*. Cambridge, MA.: MIT Press, 2014, pp. 2816–2824.
[9] J. Benamou, G. Carlier, M. Cuturi, L. Nenna, and G. Peyré, “Iterative bregman projections for regularized transportation problems,” *SIAM Journal on Scientific Computing*, vol. 37, no. 2, 2015.

[10] M. Jaggi, “Revisiting frank-wolfe: Projection-free sparse convex optimization,” in *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, GA., 2013, pp. 427–435.

[11] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, “Optimal transport for domain adaptation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, 2017.

[12] B. Kulis, K. Saenko, and T. Darrell, “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*, Colorado Springs, CO., 2011, pp. 1785–1792.

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

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

[15] H.-J. Ye, D.-C. Zhan, Y. Miao, Y. Jiang, and Z.-H. Zhou, “Rank consistency based multi-view learning: A privacy-preserving approach,” in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, Melbourne, Australia, 2015, pp. 991–1000.

[16] J. Snell, K. Swersky, and R. S. Zemel, “Prototypical networks for few-shot learning,” in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017, pp. 4080–4090.

[17] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, “Learning embedding adaptation for few-shot learning,” *CoRR*, vol. abs/1812.03664, 2018.

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

[19] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2014.