Augmented Multi-Task Learning by Optimal Transport

Boyang Liu* Pang-Ning Tan Jiayu Zhou

Abstract

Multi-task learning (MTL) provides an effective approach to improve generalization error for multiple related prediction tasks by learning the tasks jointly, assuming there is a common structure shared by their model parameters. Despite its successes, the shared parameter assumption is ineffective when the sample sizes for some tasks are too small to infer the task relationships correctly from data. To overcome this limitation, we propose a novel framework for increasing the effective sample size of each task by augmenting it with pseudo-labeled instances generated from the training data of other related tasks. Incorporating training data from other tasks is a challenge for regression problems as their data distributions may not be consistent due to the covariate shift and response drift problems. Our proposed framework addresses this challenge by coupling multi-task regression with a series of optimal transport steps to iteratively learn the pseudo-labeled instances by identifying relevant training instances from other source domains and refining the pseudo-labels until they are consistent with the training instances of the target domain. Experimental results on both synthetic and real-world data showed that our framework consistently outperformed other state-of-the-art MTL methods.

1 Introduction

The booming growth of data in recent years has made it possible to train sophisticated learning models for solving complex prediction problems. In particular, techniques such as multi-task learning (MTL) [26, 24] have been developed to address large-scale problems that can be decomposed into multiple related prediction tasks. By training the prediction models of these tasks jointly, MTL can improve generalization performance as it incorporates the task relationship information explicitly into the models, unlike single-task learning methods that build the models for each task independently.

To illustrate the advantage of using MTL, Fig. 1 shows the results of applying single-task and multi-task regression methods to a multi-region lake ecology dataset, where each task corresponds to the prediction of a nutrient variable for all the lakes located in a given region. The horizontal axis represents the percentage of labeled data used for training while the vertical axis represents the prediction error on the test sets for all regions. The figure on the left shows the results for independent lasso models (red solid line), trained to fit the data in each region separately, and the results for an MTL approach based on trace-norm regularization [15] (dashed blue line). Observe that the prediction error for MTL is an order of magnitude lower than that for independent lasso models, especially when the training set size is small. Even if 70% of the labeled data in each region are used for training, the independent lasso approach still performs poorly as some regions have too few examples to fit the local models accurately. By assuming that the model parameters share some common structure, MTL can be trained to have reasonably high accuracy even when the training set size is small [2]. However, its prediction error can still grow quite substantially as the percentage of training data decreases (see Fig. 1(right)). This is because, when the sample size of a task is too small, the parameter sharing assumption alone is insufficient to correctly learn the model parameters as the task relationships inferred from the small samples are potentially misleading [12].

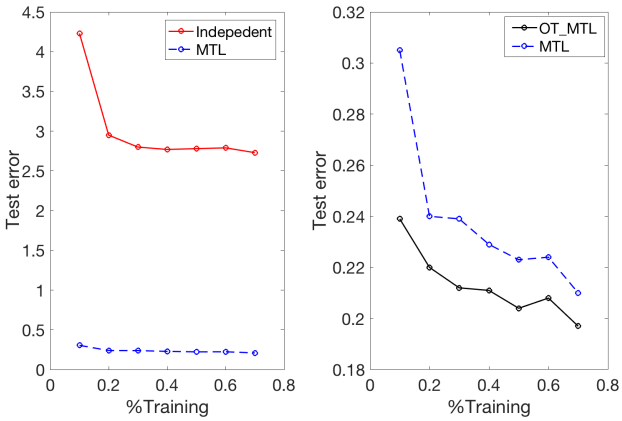
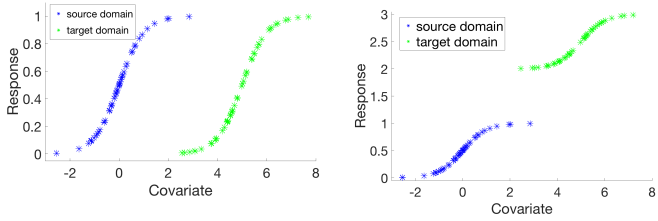


Figure 1: Comparison between the prediction errors of MTL and augmented MTL with OT against independent lasso models on a lake ecology dataset.

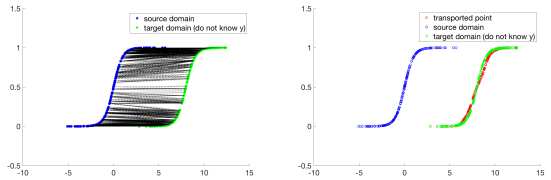
To overcome this limitation, we present an approach to increase the effective sample size for each task. A

*All authors are from the Department of Computer Science and Engineering, Michigan State University.



(a) Covariate Shift (b) Response Drift

Figure 2: The (a) covariate shift and (b) response drift problems for data distributions from different tasks.



(a) Visualization of transport matrix (b) Re-weight result

Figure 3: Optimal transport method for domain adaptation assuming there is no response drift.

naïve way to do this is by incorporating the labeled instances from all other tasks. However, this is equivalent to fitting a global model to all the tasks and is ineffective when there are significant discrepancies in the distributions of the predictor and response variables for different tasks, as shown in Figure 2. We termed the differences in distribution of predictor variables as *covariate shifts* while those due to response variables as *response drifts*.

An alternative approach is to augment the training set of each task with carefully chosen labeled instances from other related tasks. This strategy has been previously studied in the area of instance-based domain adaptation (DA) [9]. Instance-based DA methods utilize the similarity between samples to determine the importance of labeled instances from other domains. To date, most of the DA methods have been developed for classification problems, where the classes in the source and target domains have large overlap between them. For regression analysis, which is the focus of this study, instance-based DA is a harder problem as the distribution of the response variable for different tasks may vary due to the response drift problem illustrated in Figure 2(b). Therefore, the augmented instances from other domains must be properly calibrated or bias-corrected before they can be effectively used in a multi-task regression setting.

In this paper, we consider a third strategy to increase the effective sample size for MTL by assigning pseudo-labels to the unlabeled (test) instances of a given task based on its similarity to the training instances from other related tasks. These pseudo-labeled instances are then combined with the original labeled instances to train MTL models. This strategy is somewhat similar to a semi-supervised learning approach, except the pseudo-labels are determined based on the training instances from other tasks. The key challenge is to learn how to map the unlabeled (test) instances of a given task (*target domain*) to the labeled (training) instances from other related tasks (*source domains*). Towards this end, we leverage ideas from optimal transport (OT) method [7, 5] to learn an appropriate mapping (in the form of a transportation matrix) between the labeled instances of the source domain to the target domain, as shown in Fig. 3. However, conventional OT approaches cannot handle the response drift problem since it estimates the response value of each unlabeled instance in the target domain as a convex combination of the response values of the training instances in the source domain. Thus, if the range of response values in the target domain lies outside the range of the source domains (see Fig. 2(b)), the augmented instances will not be able to improve prediction in the target domain. In fact, they may degrade the overall performance as the pseudo-labels have a very different distribution than the distribution of training instances in the target domain.

To overcome this challenge, we present an optimal transport augmented multi-task learning (OT_MTL) framework to address both the covariance shift and response drift problems. Unlike conventional OT methods, our framework assumes that the pseudo-labels generated from the response values of training instances in other related tasks are biased, and thus, must be calibrated before they can be augmented with the training data. The calibration step is achieved by performing a series of OT steps to match the distribution of the pseudo-labeled instances against the distribution of the training instances in the target domain. After calibration, the pseudo-labeled instances can then be combined with the training instances to jointly build MTL models for all the tasks. As shown in Figure 1(right), OT_MTL can boost the performance of regular MTL especially when there are very few training instances available for each task. We have performed experiments using synthetic data to show the effectiveness of the approach in dealing with translational and rotational drifts in the response variable, both of which cannot be addressed by conventional OT approaches [5]. Experimental results using real-world data from various domains further validated the efficacy of the framework.

2 Related Work

Multi-Task Learning (MTL) is designed to solve multiple related learning tasks by enforcing parameter sharing across the different tasks [26]. For example, the joint feature selection approach assumes that the different tasks share the same set of discriminative features [13] by employing a group sparsity penalty [23]. Another approach assumes that the model parameters share a common, low-rank subspace, which can be found by adding a trace norm regularization penalty into the MTL formulation [4]. Other approaches include clustered MTL [25], and structured sparsity [10] methods. All of these approaches perform knowledge transfer in the parameter space only, without using training data from other related tasks, unlike the framework proposed in this paper.

Optimal Transport (OT) theory provides a systematic approach for comparing two probability distributions by seeking the least costly way to reshape one distribution into another while incorporating their geometric information. The distance given by optimized OT is called earth mover distance, which is also known as Wasserstein distance. Since the original OT problem is NP-hard [20], it is subsequently relaxed to the Monge-Kantorovich problem [11]. The Sinkhorn algorithm [8] is a popular method to accelerate the OT computation by introducing an entropy smoothing term. The algorithm has been shown to be equivalent to performing iterative Bregman projections with polytopes constraints [3]. More recently, Bregman ADMM [21] has also been proposed to efficiently solve the OT problem without entropy regularization [22]. Domain adaptation (DA) is a rich application area for OT, in which the adaptation process between the source and target domains can be viewed as an OT process [7, 6].

3 Background

This section formalizes the learning problem and reviews the MTL and OT approaches.

3.1 Problem Statement Let $\mathcal{D} = \{\mathbf{X}_i, \mathbf{y}_i\}_{i=1}^r$ be a dataset for r related learning tasks, where $\mathbf{X}_i \in \mathbb{R}^{n_i \times d}$, $\mathbf{y}_i \in \mathbb{R}^{n_i \times 1}$, and n_i is the sample size associated with the i^{th} learning task, and d is the number of features. Our goal is to construct r predictive models, $f_1, f_2, \cdots, f_r$, for the r tasks, where each model f_i maps the input features associated with instances of the i^{th} task to their corresponding response values, $\mathbf{y}_i = f_i(\mathbf{X}_i) + \epsilon_i$. We focus on linear models in this study, though the proposed approach can be readily adopted to nonlinear models as well.

3.2 Multi-task learning (MTL) The MTL problem can be generally cast into the following convex optimization problem:

$$\min_{f_1, f_2 \cdots f_r \in \Gamma} \sum_{i=1}^{r} \mathcal{L}\Big[f_i(\mathbf{X}_i; \mathbf{w}_i), \mathbf{y}_i\Big], \tag{3.1}$$

where Γ is the constraint specifying the feasible region (task relatedness structure) of the parameter space, $\mathcal{L}[\cdot]$ is the loss function, and $\mathbf{w}_i$ is the model parameter associated with task i. When Γ and the loss function are both convex, an iterative gradient descent algorithm can be used to solve Eq. 3.1. The gradient update formula can be written as follows [14]:

$$\mathbf{w}_i^{k+1} = P_\Gamma(\mathbf{w}_i^k - \alpha g(\mathbf{w}_i^k)), \tag{3.2}$$

where α is the learning rate, g is the gradient function, and $P_\Gamma(\hat{\mathbf{w}})$ is a projection operator for mapping $\hat{\mathbf{w}}$ onto the constraint space Γ.

3.3 Optimal Transport (OT) The OT approach can be used to learn a transport map $\mathbf{T}$ from the source domain Ω_s to the target domain Ω_t. Let $(\mathbf{X}^s, \mathbf{y}^s)$ denote the source domain data and $(\mathbf{X}^t, \mathbf{y}^t)$ denote the target domain data. The goal of OT is to learn $\mathbf{T}$ via the least effort principle to transform the probability distribution of instances in Ω_s to the distribution in Ω_t [7]. Let μ_s and μ_t be the empirical marginal distributions for the source and target domains, respectively:

$$\mu_s = \sum_{i=1}^{n_s} p_i^s \delta_{x_i^s}, \quad \mu_t = \sum_{i=1}^{n_t} p_i^t \delta_{x_i^t},$$

where δ_s is the Dirac delta function at $s \in \mathbb{R}^d$, whereas p_i^s and p_i^t are probability measures such that $\sum_{i=1}^{n_s} p_i^s = \sum_{i=1}^{n_t} p_i^t = 1$. OT can be cast as the following optimization problem:

$$\mathbf{T} = \underset{\mathbf{T} \in \prod(\mu_s, \mu_t)}{\arg\min} \int d(\mathbf{x}^s, \mathbf{x}^t) dT(\mathbf{x}^s, \mathbf{x}^t),$$

where $\mathbf{T}$ is a transport map from μ_s to μ_t, $d(\mathbf{x}^s, \mathbf{x}^t)$ is the distance between $\mathbf{x}^s$ and $\mathbf{x}^t$, and $\prod(\mu_s, \mu_t)$ is the probabilistic coupling between the source and target marginal distributions. For the discrete case, the above formulation can be simplified as follows [7]:

$$\begin{aligned} \mathbf{T}^* &= \arg\min_{\mathbf{T}} < \mathbf{T}, \mathbf{C} >, \\ \text{s.t. } & \mathbf{T}\mathbf{1} = \mu_s, \quad \mathbf{T}^{'}\mathbf{1} = \mu_t, \end{aligned} \tag{3.3}$$

where $\mathbf{T} \in \mathbb{R}^{n_s \times n_t}$ is the transport matrix which we need to solve, and $\mathbf{C} \in \mathbb{R}^{n_s \times n_t}$ is the transport cost matrix, which is usually chosen to be the Euclidean distance between features in the source and target domains. The Wasserstein distance between the

marginal distribution $P(\mathbf{X^s})$ and $P(\mathbf{X^t})$ is defined as $W(\mathbf{X}^s, \mathbf{X}^t) =< \mathbf{T}^*, \mathbf{C} >$. After solving for $\mathbf{T}^*$, under the assumption of uniform distribution of source and target instances as well as applying Wasserstein barycenter mapping (c.f. Eqs. (14, 15) in [7]), we have:

$$\hat{\mathbf{X}}^s = n_s \mathbf{T} \mathbf{X}^t, \quad \hat{\mathbf{X}}^t = n_t \mathbf{T}^{'} \mathbf{X}^s. \tag{3.4}$$

Note that each row in $\hat{\mathbf{X}}^s$ corresponds to a transported instance from the source to the target domains. Using $\hat{\mathbf{X}}^s$ and their corresponding $\mathbf{y}^s$, a prediction model can be trained on these instances. As shown in [7], by aligning the marginal probabilities, the transport map can address the covariate shift problem in domain adaptation. Unfortunately, for regression problems, it cannot handle the response shift problem, as illustrated in Figs. 2(b) and 3.

3.4 Sinkhorn Algorithm Solving the transportation map $\mathbf{T}$ in Eq. (3.3) is a linear programming problem, whose solution requires $O((n_s + n_t) n_s n_t \log(n_s + n_t))$ [1]. Furthermore, it has no unique solution due to the nature of the polytope constraints. The Sinkhorn algorithm with entropy regularization helps to relax the original OT problem [8] into a strongly convex problem. Entropy regularization is defined as $H(\mathbf{T}) = - < \mathbf{T}, \log \mathbf{T} >$, and the relaxed OT is given by:

$$\begin{aligned} &\mathbf{T}^* = \arg\min_{\mathbf{T}} < \mathbf{T}, \mathbf{C} > -\gamma H(\mathbf{T}), \\ &s.t.\ \mathbf{T1} = \mu_s, \mathbf{T}^{'}\mathbf{1} = \mu_t. \end{aligned} \tag{3.5}$$

The relaxed OT problem can be solved using the iterative Bregman projection [3, 19] method, where the dual projection can be efficiently implemented using the Sinkhorn-Knopp matrix scaling algorithm [16]. Sinkhorn algorithm solves the relaxed OT through the dual form of Bregman iterations:

PROPOSITION 1. (SINKHORN ALGORITHM [8]) *Let* $\xi = \exp(-\frac{\mathbf{C}}{\gamma})$, *with the initialization of* $\mathbf{v}^{(0)} = \mathbf{1}$, *the following iterations will converge to the solution of Eq. 3.5:*

$$\begin{aligned} &\mathbf{u}^{(n)} = \frac{\mu_s}{\xi \mathbf{v}^{(n)}}, \quad \mathbf{v}^{(n)} = \frac{\mu_t}{\xi^T \mathbf{u}^{(n)}} \\ &\mathbf{T}^{(n)} = \mathit{diag}(\mathbf{u}^{(n)}) \xi \mathit{diag}(\mathbf{v}^{(n)}) \end{aligned}$$

4 Proposed OT_MTL Framework

The framework proposed in this paper is designed to address the limitation of MTL when the sample sizes for some tasks are too small. For small sample sizes, the model parameters and task relationships inferred from the data have high uncertainties, and thus, are unreliable. Towards this end, we present an approach to increase the effective sample size for each task to ensure the gradient calculation using Eq. (3.2) is more stable and accurate. Specifically, we use the OT approach to generate pseudo-labeled instances for each task, which can be combined with existing labeled instances to train a more robust model. Note that each pseudo-labeled instance corresponds to an unlabeled (test) instance of a given task, whose value is estimated using the response values of the training instances from other related tasks. Our framework differs from conventional OT for domain adaptation (DA), which have mostly focused on classification problems [7, 5] and are not designed to handle the response drift problem encountered in many real-world applications.

Our proposed framework called OT_MTL performs the following two steps:

- It uses an iterative OT process to create pseudo-labeled instances for each task.
- It applies MTL to the augmented training set and jointly trains the predictive models for each task through their shared parameter regularization.

4.1 Pseudo Label Generation with Iterative OT Consider a multi-task regression problem with r tasks. For the i^{th} task, let $\mathcal{D}_i^{trn} = \{\mathbf{X}_i^{trn}, \mathbf{y}_i^{trn}\}$ be its training data and $\mathcal{D}_i^{tst} = \{\mathbf{X}_i^{tst}, \mathbf{y}_i^{tst}\}$ be its test data. Our objective is to augment the training data for each task with pseudo-labels assigned to the unlabeled (test) instances, similar to a semi-supervised learning approach. Specifically, each pseudo-labeled instance corresponds to a pair $(\mathbf{x}_{ij}, \hat{\mathbf{y}}_{ij})$, where $\mathbf{x}_{ij}$ is the unlabeled instance in test data (i.e., a row in $\mathbf{X}_i^{tst}$) and $\hat{\mathbf{y}}_{ij}$ is the estimated response value using OT. Note that the pseudo-labels $\hat{\mathbf{y}}_{ij}$ may not be entirely consistent with the true response values $\mathbf{y_{ij}}$ of the test data since they are estimated using training data from the current task as well as other related tasks.

As previously noted, a major problem in applying OT to regression problems is its inability to handle the response drift problem since the pseudo-labels are generated based on a convex combination of the response values in the source domain. To overcome this challenge, we perform an iterative series of OT steps to incrementally update the pseudo-labels of the unlabeled instances of each task (target domain). Figure 4 depicts the results of applying iterative OT steps on the sigmoid data (with both covariate shift and response drift) shown in Fig. 2(b). Initially, we apply the conventional OT method to estimate the pseudo-labels of the unlabeled (test) instances of the target domain based on the response values of training instances from other related tasks (source domains). The estimated pseudo-

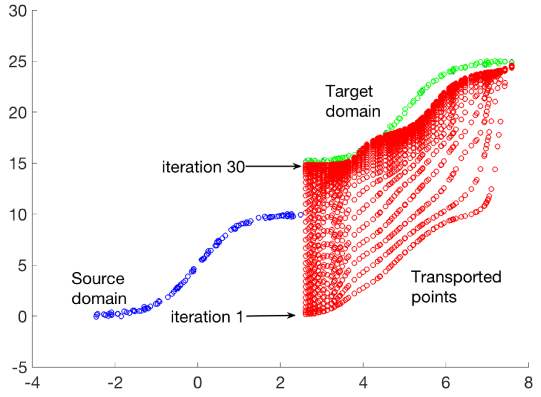


Figure 4: An illustration of the iterative OT steps.

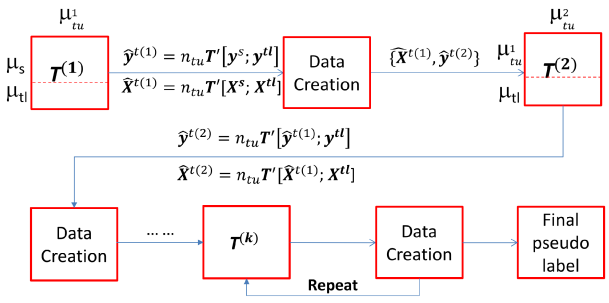


Figure 5: The source update iterations to deal with the response drift in optimal transport.

labels are shown by the sigmoidal curve for iteration 1 in the diagram. Although the results for the first iteration showed that OT can resolve the covariate shift problem, the distribution of the response values of the pseudo-labeled instances are very different than the distribution of the training instances in the target domain (shown by the green points). Our subsequent OT steps are performed to debias the original pseudo-labels and bring the distribution closer to the response value distribution of the training instances.

In conventional OT, the rows of the transport matrix $\mathbf{T}$ refer to labeled instances from the source domain whereas the columns refer to unlabeled instances from the target domain. For OT_MTL, instances that form the rows and columns of the transport matrix vary from one iteration to another. To avoid confusion, we refer to the row elements of $\mathbf{T}$ as *row instances* and the column elements as *column instances*.

Initialization: Let i be the current task for which pseudo-labels are needed for its unlabeled instances. In the first iteration, the row instances $\mathcal{D}^s$ correspond to labeled training instances from all tasks (including the target domain) whereas the column instances $\mathcal{D}^t$ correspond to unlabeled instances from the target domain.

- Row instances: $\mathcal{D}_1^l \bigcup \mathcal{D}_2^l \cdots \bigcup \mathcal{D}_r^l \equiv (\mathbf{X}^s, \mathbf{y}^s)$
- Column instances: $\mathcal{D}_i^u \equiv (\mathbf{X}_i^{tu})$.

where each $\mathcal{D}_j^l = (\mathbf{X}_j^l, \mathbf{y}_k^l)$ denotes the labeled (training) instances from the j^{th} task and $\mathcal{D}_i^u = \mathbf{X}_i^{tu}$ denotes the unlabeled instances from the i^{th} task whose pseudo-labels are to be estimated. The marginal distributions of the row and column instances are as follows:

$$\mu_s = \sum_{i=1}^{N_s} p_i^s \delta_{x_i^s}, \quad \mu_t = \sum_{i=1}^{n_i^u} p_i^t \delta_{x_i^{tu}}, \tag{4.6}$$

where $N_s = \sum_i n_i^l$, whereas p_i^s and p_i^t are probability measures such that $\sum_{i=1}^{n_s} p_i^s = \sum_{i=1}^{n_t} p_i^t = 1$. We then compute the cost matrix $\mathbf{C}$ between the row and column instances based on the Euclidean distance of their predictor variables, i.e., $\mathbf{C}_{ij} = \|\mathbf{X}_i^s - \mathbf{X}_j^t\|$. Given the cost matrix and the marginal distributions, we then apply the Sinkhorn algorithm to learn the transport map $\mathbf{T}$ (see Eq. 3.5). The pseudo labels of the column instances in $\mathbf{T}$ are then given by

$$\hat{\mathbf{y}}_{\text{pseudo}}^{t(1)} = n_{tu} \mathbf{T}' \mathbf{z}, \tag{4.7}$$

where n_{tu} is the number of unlabeled instances and $\mathbf{z} = [\mathbf{y}_1^l; \mathbf{y}_2^l; \ldots; \mathbf{y}_r^l]$ is the response values of the column instances. The superscript (1) in $\hat{\mathbf{y}}^{t(1)}$ denote the pseudo-labels after the first iteration. $\mathbf{T}$ is also used to generate pseudo-covariates as follows:

$$\hat{\mathbf{X}}_{\text{pseudo}}^{tu(1)} = n_{tu} \mathbf{T}' [\mathbf{X}^s; \mathbf{X}^{tl}] \tag{4.8}$$

Both $\hat{\mathbf{X}}_{\text{pseudo}}^{tu(1)}$ and $\hat{\mathbf{y}}_{\text{pseudo}}^{t(1)}$ will be used in subsequent OT steps. The procedure for the first OT step is summarized in the top half of Figure 5.

k-th Iteration. In subsequent iterations, the pseudo-labeled instances generated from previous iterations and the training instances of the target domain will form the row instances, while the column instances are unchanged:

- Row instances: $\hat{\mathcal{D}}_{\text{pseudo}}^{(k-1)} \bigcup \mathcal{D}_i^l \equiv (\mathbf{X}^{tl}, \mathbf{y}^{tl})$
- Column instances: $\mathcal{D}_i^u \equiv (\mathbf{X}_i^{tu})$.

where $\hat{\mathcal{D}}_{\text{pseudo}}^{(k-1)} = (\hat{\mathbf{X}}_{\text{pseudo}}^{tu(k-1)}, \hat{\mathbf{y}}_{\text{pseudo}}^{tu(k-1)})$ corresponds to the pseudo-labeled instances generated in the previous iteration. This procedure is repeated until the stopping criteria (to be discussed below) is met. This iterative procedure is shown in the bottom half diagram of Fig. 5. The rationale for our iterative approach is as follows. From Eq. 4.7, observe that the pseudo-labels $\hat{\mathbf{y}}_{\text{pseudo}}^{t}$ depends on $\mathbf{z}$. Since not all labeled instances from

other tasks are related to the target task, it is necessary to increase the weight of the labeled instances from the target task and decrease the weight of unrelated labeled instances from other tasks. Our iterative OT strategy would gradually increase the weights of labeled instances in the target domain, thus allowing it to handle the drift response problem. Formally, the pseudo label iteration for task i can be stated as follows:

$$\mathbf{C}^{(\mathbf{k+1})} = d([\hat{\mathbf{X}}_{i,\text{pseudo}}^{tu(k)}, \mathbf{X}_i^{tl}], \mathbf{X}_i^{tu})$$

$$\mu_{(k+1)} = \sum_{i=1}^{n_i^l+n_i^t} p_i^s \delta_{[\hat{x}_{i,\text{pseudo}}^{tu(k)}; x_i^{tl}]}$$

$$\mathbf{T}^{(k+1)} = \underset{\mathbf{T}\in\prod(\mu_s^{k+1},\mu_t)}{\arg\min} < \mathbf{T}, \mathbf{C}^{(\mathbf{k+1})} > + \gamma H(\mathbf{T})$$

$$\hat{\mathbf{y}}_{i,\text{pseudo}}^{tu(k+1)} = n_i^{tu}\mathbf{T}^{(k+1)'}[\hat{\mathbf{y}}_{i,\text{pseudo}}^{tu(k)}; \mathbf{y}_i^l]$$

$$\hat{\mathbf{X}}_{i,\text{pseudo}}^{tu(k+1)} = n_i^{tu}\mathbf{T}^{(k+1)'}[\hat{\mathbf{X}}_{i,\text{pseudo}}^{tu(k)}; \mathbf{X}_i^{tl}], \tag{4.9}$$

Stopping Criteria. A stopping criteria is needed to prevent the pseudo-labeled instances from overfitting the training instances of the target domain. We consider the following stopping criteria: Let $\hat{\mathbf{y}}_{i,\text{pseudo}}^{tu(k)}$ be the pseudo-labels generated after k iterations for i^{th} task (domain) and $\mathbf{y}_i^l$ be the response values of the labeled instances in the target domain for i^{th} task. Given a threshold τ, the iterative OT-steps for i^{th} task will terminate if

$$|\text{mean}(\hat{\mathbf{y}}_{i,\text{pseudo}}^{tu(k)}) - \text{mean}(\mathbf{y}_i^l)| < \tau$$

Overall, our iterative OT algorithm is summarized in Algorithm 4.1. We named the algorithm **OT_MTL** as it is an augmentation of OT into MTL framework.

4.2 MTL with Pseudo Label After generating the pseudo labels, we can apply MTL on both the labeled and pseudo-labeled instances to train the prediction models for each task as follows:

$$\underset{f_1,f_2\ldots f_r}{\arg\min} \sum_{i=1}^{r} ||f_i\left([\mathbf{X}_i^l; \mathbf{X}_i^{tu}], \mathbf{w}_i\right) - [\mathbf{y}_i^l; \hat{\mathbf{y}}_{i,\text{pseudo}}^{tu}]||^2 + \Omega(f). \tag{4.10}$$

The MTL approach used depends on the choice of regularization penalty Ω [26]. By incorporating the pseudo-labeled instances, this increases the effective sample size for each task, and thus, is expected to improve the generalization performance of the models. Eq. (4.10) can be solved by iteratively applying the projected gradient descent algorithm [26].

5 Experimental Evaluation

5.1 Data We use both synthetic and real-world data to evaluate the performance of our algorithm. A summary of the real-world data is given in Table 1.

Algorithm 1 OT_MTL

Input: $\mathcal{D} = \{\mathbf{X}_i^l, \mathbf{y}_i^l, \mathbf{X}_i^{tu}\}_{i=1}^r$, maxiter, γ, τ
Output: The different functions $f_1, f_2 \ldots f_r$
begin
  Concatenate all $\{\mathbf{y}_i^l\}_{i=1}^r$ for every task as $\mathbf{z}$.
  Concatenate all $\{\mathbf{X}_i^l\}_{i=1}^r$ for every task as $\mathbf{h}$.
  for each task i
    $\mathbf{C} = d(\mathbf{h}, \mathbf{X}_i^{tu})$
    $\mathbf{T} = Sinkhorn(\mu_s, \mu_t, \mathbf{C})$
    $\hat{\mathbf{X}}_{i,\text{pseudo}}^{tu} = n_i^{tu}\mathbf{T}'\mathbf{h}$
    $\hat{\mathbf{y}}_{i,\text{pseudo}}^{tu} = n_i^{tu}\mathbf{T}'\mathbf{z}$
    for j = 1 **to** maxiter
      if $|mean(\mathbf{y}_i^l) - mean(\hat{\mathbf{y}}_{i,\text{pseudo}}^{tu})| >= \tau$
        Update $\mathbf{C}, \mu_\mathbf{s}, \mathbf{T}, \hat{\mathbf{y}}_{i,\text{pseudo}}^{tu}, \hat{\mathbf{X}}_{i,\text{pseudo}}^{tu}$ with 4.9
      else
        break
      end
    end
  end
  Solve Eq. (4.10) to learn the functions $f_1, f_2 \ldots f_r$.
end

5.1.1 Synthetic Data The purpose of using synthetic data is to illustrate the response drift problem in multi-task regression and how our iterative OT approach can address this problem. For brevity, we set the number of tasks equals to two and generate two synthetic data with different types of response drifts. The first dataset captures a response drift due to translation while the second dataset captures a response drift due to rotation. We also set the number of features in these datasets to be one to enable better visualization of the results. For each dataset, we generate 100 instances for the source and target domains. The response values for all instances in the source domain are assumed to be known. For the target domain, only ten instances are assumed to be labeled (i.e., have known response values) while the rest are assumed to be unlabeled when applying the OT_MTL algorithm.

5.1.2 Lake Ecology Data [17] Each data instance corresponds to a lake. There are 13 predictors and 4 response variables associated with each lake—total phosphorus (tp), total nitrogen (tn), chlorophyll-a(chla), and Secchi depth (secchi). Each predictor variable is standardized to have zero mean and unit standard deviation while the response variable is log transformed to avoid skewness in the data distribution. The lakes are grouped into regions and each region is treated as a separate task.

5.1.3 School Data This dataset contains the test scores of 15,362 students from 139 schools provided by

Table 1: Summary statistics for our experiment dataset

Response	# tasks	# instances	# instances/tasks
TP	86	4352	1-369
TN	83	1946	1-236
Chla	87	5592	1-575
Secchi	88	5796	1-583
School	139	15362	22-251

the Inner London Education Authority(ILEA). Each school is treated as a separate learning task and our goal is to predict the exam scores using 28 features. All features are standardized to have zero mean and unit standard deviation.

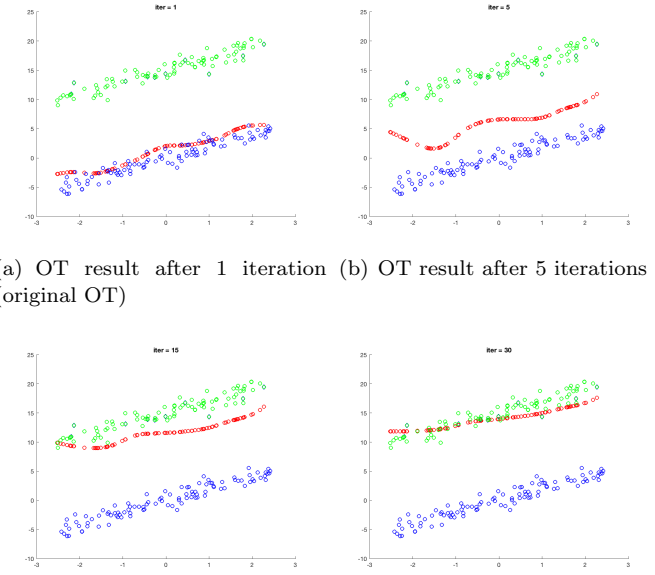
6 Experimental Setup

We have used the following baseline algorithms for comparison purposes:

- **Global_L1**: A global lasso regression model trained using labeled data from all the tasks [18].
- **Independent_L1**: An independent lasso model is separately trained for each task [18].
- **Least_L21**: An MTL approach based on L21 norm with group sparsity assumption [13].
- **Least_Lasso**: MTL based on L1 regularization.
- **Least_Trace**: An MTL approach based on nuclear norm regularization for low rank assumption [26].
- **OT**: Conventional OT method [8] which uses the pseudo labels of the unlabeled instances as its predicted response values.
- **JDOT**: This corresponds to applying the joint distribution optimal transport (JDOT) algorithm [5] to obtain the pseudo labels, followed by a MTL approach with nuclear norm regularization.

For fair comparison, our proposed OT_MTL framework also uses the nuclear norm regularization to train its models. We also consider a variation of our approach, called **OT_MTL_init**, which terminates after 1 iteration. This approach does not perform iterative update to resolve the response drift problem and is quite similar to the MDOT algorithm [7] except the pseudo-labels are created for unlabeled instances in the target domain instead of the transported instances from source domain.

We use root mean square error as our evaluation metric. The metric is computed by concatenating the predicted values of the test instances from all the tasks. We perform nested cross-validation for hyper-parameter tuning and model evaluation. For all the baseline as well as our algorithm, the hyperparameters are chosen from the same set. The stopping threshold τ for OT_MTL is chosen to be 0.005 for lake ecology data and 3 for the school data. For **JDOT**, the trade-off parameter α is chosen to be $\frac{1}{max(\mathbf{C}_{i,j})}$, which was suggested by the authors in [5].



(a) OT result after 1 iteration (b) OT result after 5 iterations (original OT)

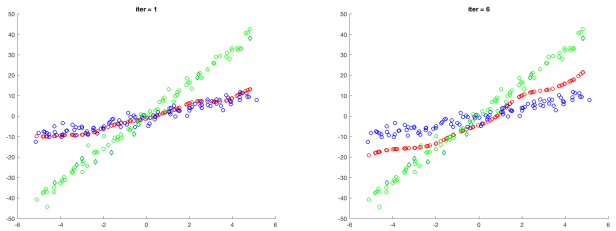
(c) OT result after 15 iterations (d) OT result after 30 iterations

Figure 6: Application of OT_MTL to synthetic data with translated response shift. Instances in the source domain are represented as blue dots while those in the target domain are represented as green dots (if unlabeled) or blue diamonds (if labeled). The red dots are pseudo labeled instances generated by OT_MTL.

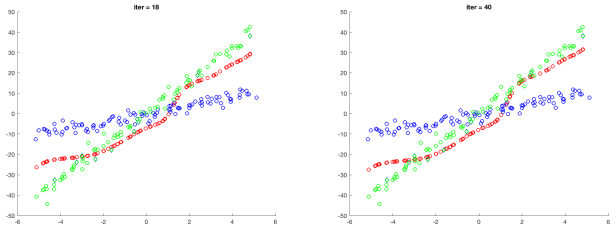
7 Results on Synthetic Data

In this section, we demonstrate how the OT_MTL algorithm deals with different types of response drifts. First, we consider the case in which the response values of the target domain are vertically shifted compared to the response values of the source domain. Figure 6 shows the pseudo-labeled instances generated by OT_MTL from 1 to 30 iterations. Instances from the source domain are shown in blue while those from the target domain are shown in green. Even though the true response value for all target instances are shown in the diagram, only 10% of them are used for training. After 1 iteration, the pseudo-labeled instances do not reflect the true distribution of the target domain due to the response drift problem. With increasing number of iterations, the pseudo-labeled instances (shown as red points) converge closer to their true distribution.

Next, we consider the effect of "rotated" response drift, where the relationship between the predictor and response variables in the target domain is a rotated version of the relationship in the source domain, as shown in Fig. 7. With increasing number of iterations, the results show that our algorithm continuously learn the shape of the rotated space. After 40 iterations, the pseudo-labeled instances (shown as red points) are close to the true labeled instances of the rotated target



(a) OT result after 1 iteration (b) OT result after 6 iterations (original OT)



(c) OT result after 18 iterations (d) OT result after 40 iterations

Figure 7: Application of OT_MTL to synthetic data with rotated response shift.

domain (shown as green points).

8 Results on Real-World Data

We now consider the performance comparison of the various algorithms on the lake ecology and school data. For lake ecology, there are 4 response variables considered: TP, TN, Chla, and Secchi. The results are summarized in Table 2. The results show that our proposed framework, OT_MTL, consistently outperforms all other methods on all datasets. It also outperforms OT_MTL_init, which suggests the presence of response drifts that may degrade the overall performance of JDOT and OT_MTL_init.

We further investigate the performance improvement of OT_MTL against OT_MTL_init for varying training set sizes (from 10% to 50%). We define performance improvement in terms of the following metric:

$$\text{(8.11)} \quad \text{Improv.} = \frac{\text{rmse(baseline)} - \text{rmse(OT\_MTL)}}{\text{rmse(baseline)}}$$

The results shown in Fig.8 suggest that OT_MTL achieves performance improvement close to 10% or more on the lake ecology datasets, with larger improvements observed when the training set size is small. For school data, since the sample size is already large, our approach only improves little compares to the lake ecology datasets.

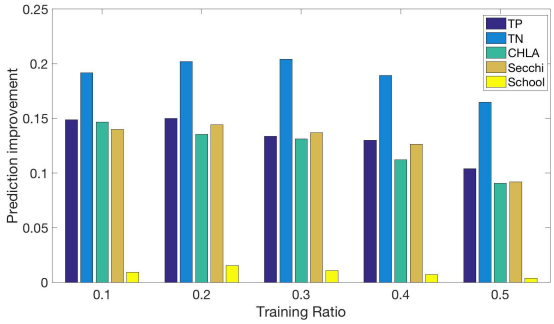


Figure 8: Comparison between OT_MTL and OT_MTL_init. The x-axis represents the training ratio while the y-axis represents the prediction improvement metric defined in Eq. (8.11).

9 Conclusion

In this paper, we present a novel method called OT_MTL that combines optimal transport with multi-task learning to address small sample size and response drift problems in regression. OT_MTL employs both parameter sharing and sample sharing strategies to enhance its generalization performance. Unlike existing OT methods for domain adaptation, our method employs an iterative source update approach to overcome the response drift problem. Experimental results on both synthetic and real-world datasets validate the effectiveness of our method.

10 Acknowledgement

This research was supported in part by the NSF under grant (EF-1638679, IIS-1615612, IIS-1615597, and IIS-1749940) and ONR under grand (N00014-17-1-2265). Any use of trade, firm or product names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

References

[1] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. Network flows. 1988.

[2] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in neural information processing systems*, pages 41–48, 2007.

[3] Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

[4] Jianhui Chen, Jiayu Zhou, and Jieping Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *Proceedings of the 17th ACM*

Table 2: RMSE Comparison for 5 prediction task in lake ecology data (TP,TN,Chla,Secchi) and school data when the training set ratio is 0.3.

method	TP	TN	Chla	Secchi	School
Global_L1	1.368 ± 0.004	2.731 ± 0.003	1.010 ± 0.004	0.453 ± 0.002	1.836 ± 0.004
Independent_L1	1.932 ± 0.736	2.663 ± 0.298	1.926 ± 2.198	0.909 ± 0.600	1.884 ± 0.007
Least_L21	0.589 ± 0.336	0.941 ± 0.593	0.939 ± 0.106	0.416 ± 0.193	0.873 ± 0.015
Least_Lasso	0.739 ± 0.197	1.297 ± 0.223	0.939 ± 0.106	0.342 ± 0.133	0.891 ± 0.033
Least_Trace	0.364 ± 0.011	0.341 ± 0.022	0.397 ± 0.004	0.250 ± 0.011	0.849 ± 0.007
OT	0.421 ± 0.002	0.331 ± 0.002	0.523 ± 0.003	0.323 ± 0.002	0.879 ± 0.002
OT_MTL_init	0.359 ± 0.002	0.284 ± 0.005	0.450 ± 0.003	0.270 ± 0.002	0.837 ± 0.001
JDOT_MTL_Trace	0.375 ± 0.015	0.300 ± 0.013	0.467 ± 0.006	0.290 ± 0.022	0.957 ± 0.005
OT_MTL	**0.311 ± 0.007**	**0.226 ± 0.003**	**0.391 ± 0.006**	**0.233 ± 0.006**	**0.828 ± 0.001**

SIGKDD international conference on Knowledge discovery and data mining, pages 42–50. ACM, 2011.

[5] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 3733–3742, 2017.

[6] Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 274–289. Springer, 2014.

[7] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 39(9):1853–1865, 2017.

[8] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[9] Hal Daume III and Daniel Marcu. Domain adaptation for statistical classifiers. *Journal of artificial Intelligence research*, 26:101–126, 2006.

[10] Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep K Ravikumar. A dirty model for multi-task learning. In *Advances in neural information processing systems*, pages 964–972, 2010.

[11] Leonid Kantorovitch. On the translocation of masses. *Management Science*, 5(1):1–4, 1958.

[12] Kaixiang Lin and Jiayu Zhou. Interactive multi-task relationship learning. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 241–250. IEEE, 2016.

[13] Feiping Nie, Heng Huang, Xiao Cai, and Chris H Ding. Efficient and robust feature selection via joint 2, 1-norms minimization. In *Advances in neural information processing systems*, pages 1813–1821, 2010.

[14] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[15] Ting Kei Pong, Paul Tseng, Shuiwang Ji, and Jieping Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20(6):3465–3489, 2010.

[16] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[17] Patricia A Soranno, Linda C Bacon, Michael Beauchene, Karen E Bednar, Edward G Bissell, Claire K Boudreau, Marvin G Boyer, Mary T Bremigan, Stephen R Carpenter, Jamie W Carr, et al. Lagosne: a multi-scaled geospatial and temporal database of lake ecological context and water quality for thousands of us lakes. *GigaScience*, 6(12):1–22, 2017.

[18] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[19] Koji Tsuda, Gunnar Rätsch, and Manfred K Warmuth. Matrix exponentiated gradient updates for online learning and bregman projection. *Journal of Machine Learning Research*, 6(Jun):995–1018, 2005.

[20] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[21] Huahua Wang and Arindam Banerjee. Bregman alternating direction method of multipliers. In *NIPS*, pages 2816–2824, 2014.

[22] Jianbo Ye, Panruo Wu, James Z Wang, and Jia Li. Fast discrete distribution clustering using wasserstein barycenter with sparse support. *IEEE Transactions on Signal Processing*, 65(9):2317–2332, 2017.

[23] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[24] Yu Zhang and Qiang Yang. A survey on multi-task learning. *CoRR*, abs/1707.08114, 2017.

[25] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Clustered multi-task learning via alternating structure optimization. In *Advances in neural information processing systems*, pages 702–710, 2011.

[26] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Malsar: Multi-task learning via structural regularization. *Arizona State University*, 21, 2011.