A Principled Approach for Learning Task Similarity in Multitask Learning

Changjian Shui[1*], **Mahdieh Abbasi**[1], **Louis-Émile Robitaille**[1], **Boyu Wang**[2], **Christian Gagné**[1]
[1]Université Laval
[2]University of Pennsylvania
{changjian.shui.1, mahdieh.abbasi.1, louis-emile.robitaille.1}@ulaval.ca, boyuwang@seas.upenn.edu, christian.gagne@gel.ulaval.ca

Abstract

Multitask learning aims at solving a set of related tasks simultaneously, by exploiting the shared knowledge for improving the performance on individual tasks. Hence, an important aspect of multitask learning is to understand the similarities within a set of tasks. Previous works have incorporated this similarity information explicitly (e.g., weighted loss for each task) or implicitly (e.g., adversarial loss for feature adaptation), to achieve good empirical performances. However, the theoretical motivations for adding task similarity knowledge are often missing or incomplete. In this paper, we give a different perspective from a theoretical point of view to understand this practice. We first provide an upper bound on the generalization error of multitask learning, showing the benefit of explicit and implicit task similarity knowledge. We systematically derive the bounds based on two distinct task similarity metrics: $\mathcal{H}$ divergence and Wasserstein distance. From these theoretical results, we revisit the Adversarial Multitask Neural Network, proposing a new training algorithm to learn the task relation coefficients and neural network parameters iteratively. We assess our new algorithm empirically on several benchmarks, showing not only that we find interesting and robust task relations, but that the proposed approach outperforms the baselines, reaffirming the benefits of theoretical insight in algorithm design.

1 Introduction

Traditional machine learning mainly focused on designing learning algorithms for individual problems. While significant progress has been achieved in applied and theoretical research, it still requires a large amount of labelled data in such a context to obtain a small generalization error. In practice, this can be highly prohibitive, e.g., for modelling users' preferences for products [Murugesan and Carbonell, 2017], for classifying multiple objects in computer vision [Long *et al.*, 2017], or for analyzing patient data in computational healthcare [Wang and Pineau, 2015]. In the multitask learning (MTL) scenario, an agent learns the shared knowledge between a set of related tasks. Under different assumptions on task relations, MTL has been shown to reduce the amount of labelled examples required per task to reach an acceptable performance.

Understanding the theoretical assumptions of the tasks relationship plays a key role in designing a good MTL algorithm. In fact, it determines which *inductive bias* should be involved in the learning procedure. Recently, there are many successful algorithms that rely on task similarity information, which assumes the *Probabilistic Lipschitzness* (PL) condition [Urner and Ben-David, 2013] as the inductive bias. For instance, [Murugesan and Carbonell, 2017; Murugesan *et al.*, 2016; Pentina and Lampert, 2017] minimize a weighted sum of empirical loss in which similar tasks are assigned higher weights. These approaches explicitly estimate the task similarities through a linear model. Since these approaches are estimated in the original input space, it is difficult to handle the *covariate shift* problem. Therefore, many neural network based approaches started to explore tasks similarities implicitly: [Liu *et al.*, 2017; Li *et al.*, 2018] use adversarial losses by feature adaptation, minimizing the distribution distance between the tasks to construct a shared feature space. Then, the hypothesis for the different tasks are learned over this adapted feature space.

The implicit similarity learning approaches are inspired from the idea of Generative Adversarial Networks (GANs) [Goodfellow *et al.*, 2014]. However, the fundamental implications of incorporating task similarity information in MTL algorithms are not clear. The two main questions are *why* should we combine explicit and implicit similarity knowledge in the MTL framework and *how* can we properly do it.

Previous works either consider explicit or implicit similarity knowledge separately, or combine them heuristically in some specific applications. In contrast, the main goal of our work is to give a rigorous analysis of the benefits of task similarities and derive an algorithm which properly uses this information. We start by deriving an upper bound on the generalization error of MTL under different similarity metrics (or adversarial loss). These bounds show the motivation behind the use of adversarial loss in MTL, that is to control the generalization error. Then, we derive a new procedure to update the relationship coefficients from these theoretical guarantees. This procedure allows us to bridge the gap between the explicit and implicit similarities, which have been previously seen as disjointed

*Contact Author

or treated heuristically. We then derive a new algorithm to train the Adversarial Multitask Neural Network (AMTNN) and validate it empirically on two benchmarks: digit recognition and Amazon sentiment analysis. The results show that our method not only highlights some interesting relations, but also outperforms the previous baselines, reaffirming the benefits of theory in algorithm design.

2 Related Work

Multitask Learning (MTL) A broad and detailed presentation of the general MTL is provided in some survey papers [Zhang and Yang, 2017; Ruder, 2017]. More specifically related to our work, we note several practical approaches that use tasks relationship to improve empirical performances: [Zhang and Yeung, 2010] solve a convex optimization problem to estimate tasks relationships, while [Long *et al.*, 2017; Kendall *et al.*, 2018] propose probabilistic models through construction of a task covariance matrix or estimate the multitask likelihood from a deep Bayes model. On the theoretical side, [Murugesan *et al.*, 2016; Murugesan and Carbonell, 2017; Pentina and Lampert, 2017] analyze the weighted sum loss algorithm and its applications in the online, active and transductive scenarios. Moreover, [Maurer *et al.*, 2016] analyze the generalization error of representation-based approaches, and [Zhang, 2015] analyze the algorithmic stability in MTL.

Similarity Metrics and Adversarial Loss The *similarity metric* (or distribution divergence) is widely used in deep generative models [Goodfellow *et al.*, 2014; Arjovsky *et al.*, 2017], domain adaptation [Ben-David *et al.*, 2010; Ganin *et al.*, 2016; Redko *et al.*, 2017], robust learning [Konstantinov and Lampert, 2019], and meta-learning [Rakotomamonjy *et al.*, 2018]. In transfer learning, adversarial losses are widely used for feature adaptation, since the transfer procedure is much more efficient on a shared representation. In applied transfer learning, $\mathcal{H}$ divergence [Ganin *et al.*, 2016] and Wasserstein distance [Li *et al.*, 2018] are widely used in the divergence metric. As for MTL applications, [Liu *et al.*, 2017] and [Kremer *et al.*, 2018] apply $\mathcal{H}$-divergence in natural language processing for text classification and speech recognition, while [Janati *et al.*, 2018] are the first to propose the use of Wasserstein distance to estimate the similarity of linear parameters instead of the data generation distributions. As for the theoretical understanding, [Lee and Raginsky, 2018] analyzes the minimax statistical property in the Wasserstein distance.

3 Preliminaries

Considering a set of T tasks $\{\hat{\mathcal{D}}_t\}_{t=1}^T$, in which the observations are generated by the underlying distribution $\mathcal{D}_t$ over $\mathcal{X}$ and the real target is determined by the underlying labelling functions $f_t : \mathcal{X} \to \mathcal{Y}$ for $\{(\mathcal{D}_t, f_t)\}_{t=1}^T$. Then, the goal of MTL is to find T hypothesis: $h_1, \ldots, h_T$ over the hypothesis space $\mathcal{H}$ to control the average expected error of all the tasks:

$$\frac{1}{T}\sum_{t=1}^{T} R_t(h_t),$$

where $R_t(h_t) \equiv R_t(h_t, f_t) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}_t}\ell(h_t(\mathbf{x}), f_t(\mathbf{x}))$ is the expected risk at task t and ℓ is the loss function. Throughout the theoretical part, the loss is $\ell(h_t(\mathbf{x}), f_t(\mathbf{x})) = |h_t(\mathbf{x}) - f_t(\mathbf{x})|$, which is coherent with [Pentina and Lampert, 2017; Li *et al.*, 2018; Ben-David *et al.*, 2010; Ganin *et al.*, 2016; Redko *et al.*, 2017]. If h, f are the binary mappings with output in $\{-1, 1\}$, it recovers the typical zero-one loss.

We also assume that each task has m_t examples, with $\sum_{t=1}^T m_t = m$ examples in total. Then for each task t, we consider a minimization of weighted empirical loss for each task. That means we define a simplex $\boldsymbol{\alpha}_t \in \Delta^T = \{\boldsymbol{\alpha}_{t,i} \geq 0,\ \sum_{i=1}^T \boldsymbol{\alpha}_{t,i} = 1\}$ for the corresponding weight for task t. Then the weighted empirical error w.r.t. the hypothesis h for task t can be written as:

$$\hat{R}_{\boldsymbol{\alpha}_t}(h) = \sum_{i=1}^{T} \boldsymbol{\alpha}_{t,i}\hat{R}_i(h),$$

where $\hat{R}_i(h) = \frac{1}{m_i}\sum_{j=1}^{m_i}\ell(h(\mathbf{x}_j), \mathbf{y}_j)$ is the average empirical error for task i.

4 Similarity Measures

As we illustrated in the previous section, we are interested in task similarity metrics in MTL. Therefore, the first element to determine is how to measure the similarity between two distributions. For this, we introduce two metrics: $\mathcal{H}$-divergence [Ben-David *et al.*, 2010] and Wasserstein distance [Arjovsky *et al.*, 2017], which are widely used in machine learning.

4.1 $\mathcal{H}$-divergence

Given an input space $\mathcal{X}$ and two probability distributions $\mathcal{D}_i$ and $\mathcal{D}_j$ over $\mathcal{X}$, let $\mathcal{H}$ be a hypothesis class on $\mathcal{X}$. We define the $\mathcal{H}$-divergence of two distributions as

$$d_{\mathcal{H}}(\mathcal{D}_i, \mathcal{D}_j) = \sup_{h,h'\in\mathcal{H}} |R_i(h, h') - R_j(h, h')|.$$

The empirical $\mathcal{H}$-divergence corresponds to:

$$d_{\mathcal{H}}(\hat{\mathcal{D}}_i, \hat{\mathcal{D}}_j) = \sup_{h,h'\in\mathcal{H}} |\hat{R}_i(h, h') - \hat{R}_j(h, h')|.$$

4.2 Wasserstein Distance

We assume $\mathcal{X}$ is the measurable space and denote $\mathcal{P}(\mathcal{X})$ as the set of all probability measures over $\mathcal{X}$. Given two probability measures $\mathcal{D}_i \in \mathcal{P}(\mathcal{X}_1)$ and $\mathcal{D}_j \in \mathcal{P}(\mathcal{X}_2)$, the *optimal transport* problem can be defined as searching for a probabilistic coupling γ refined as a joint probability measure over $\mathcal{X}_1 \times \mathcal{X}_2$ with marginals $\mathcal{D}_i$ and $\mathcal{D}_j$ for all $\mathbf{x}, \mathbf{y}$ that are minimizing the cost of transport w.r.t. some cost function:

$$\begin{aligned} &\text{argmin}_\gamma \int_{\mathcal{X}_1\times\mathcal{X}_2} c(\mathbf{x}, \mathbf{y})^p d\gamma(\mathbf{x}, \mathbf{y}), \\ &\text{s.t.} \quad \mathbf{P}^{\mathcal{X}_1}\#\gamma = \mathcal{D}_i; \quad \mathbf{P}^{\mathcal{X}_2}\#\gamma = \mathcal{D}_j, \end{aligned}$$

where $\mathbf{P}^{\mathcal{X}_1}$ is the projection over $\mathcal{X}_1$ and $\#$ denotes the push-forward measure. The Wasserstein distance of order p between $\mathcal{D}_i$ and $\mathcal{D}_j$ for any $p \geq 1$ is defined as:

$$W_p^p(\mathcal{D}_i, \mathcal{D}_j) = \inf_{\gamma\in\Pi(\mathcal{D}_i,\mathcal{D}_j)} \int_{\mathcal{X}_1\times\mathcal{X}_2} c(\mathbf{x}, \mathbf{y})^p d\gamma(\mathbf{x}, \mathbf{y}),$$

where $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ is the cost function of transportation of one unit of mass $\mathbf{x}$ to $\mathbf{y}$ and $\Pi(\mathcal{D}_i, \mathcal{D}_j)$ is the collection of all joint probability measures on $\mathcal{X} \times \mathcal{X}$ with marginals $\mathcal{D}_i$ and $\mathcal{D}_j$. Throughout this paper, we only consider the case of $p = 1$, i.e., Wasserstein-1 distance.

5 Theoretical Guarantees

Based on the definitions of the distribution similarity metric, we are demonstrating that the generalization error in the multitask learning can be upper bounded by the following result:

Theorem 1. *Let $\mathcal{H}$ be a hypothesis family with a VC-dimension d. If we have T tasks generated by the underlying distribution and labelling function $\{(\mathcal{D}_1, f_1), \ldots, (\mathcal{D}_T, f_T)\}$ with observation numbers $m_1, \ldots, m_T$. If we adopt the $\mathcal{H}$ divergence as a similarity metric, then for any simplex $\boldsymbol{\alpha}_t \in \Delta^T$, and for $\delta \in (0, 1)$, with probability at least $1 - \delta$, for $h_1, \ldots, h_T \in \mathcal{H}$, we have:*

$$\frac{1}{T}\sum_{t=1}^{T} R_t(h_t) \leq \underbrace{\frac{1}{T}\sum_{t=1}^{T} \hat{R}_{\boldsymbol{\alpha}_t}(h_t)}_{\text{Weighted empirical loss}} + \underbrace{C_1 \sum_{t=1}^{T}\Big(\sqrt{\sum_{i=1}^{T}\frac{\boldsymbol{\alpha}_{t,i}^2}{\beta_i}}\Big)}_{\text{Coefficient regularization}}$$
$$+ \underbrace{\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{T}\boldsymbol{\alpha}_{t,i} d_{\mathcal{H}}(\hat{\mathcal{D}}_t, \hat{\mathcal{D}}_i)}_{\text{Empirical distribution distance}} + \underbrace{C_2 + \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{T}\boldsymbol{\alpha}_{t,i}\lambda_{t,i}}_{\text{Complexity \& optimal expected loss}},$$

where $\beta_i = \frac{m_i}{m}$, $C_1 = 2\sqrt{\frac{2(d\log(\frac{2em}{d}) + \log(\frac{16T}{\delta}))}{m}}$ and $C_2 = 2\min_{i,j}\sqrt{\frac{2d\log(2m_{i,j}) + \log(32T/\delta)}{m_{i,j}}}$ with $m_{i,j} = \min\{m_i, m_j\}$ and $\lambda_{i,j} = \inf_{h\in\mathcal{H}}\{R_i(h) + R_j(h)\}$ (joint expected minimal error w.r.t. $\mathcal{H}$).

The details of the proof are provided in the arXiv version of the paper[1]. Theorem 1 illustrates that the upper bound on the generalization error in our MTL settings can be decomposed into the following terms:

- The empirical loss and empirical distribution similarities control the weights (or task relation coefficient) $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T$. For instance, for a given task t, if task i has a small empirical distance $d_{\mathcal{H}}(\hat{\mathcal{D}}_t, \hat{\mathcal{D}}_i)$ and hypothesis h_t has a small empirical loss $\hat{R}_i(h_t)$ on task i, it means that task i is very similar to t. Hence, more information should be borrowed from task i when learning t and the corresponding coefficient $\boldsymbol{\alpha}_{t,i}$ should have high values.
- Simultaneously, the *coefficient regularization term* prevents the relation coefficients locating only on the $\boldsymbol{\alpha}_{t,t}$, in which it will completely recover the independent MTL framework. Then the coefficient regularization term proposed a trade-off between learning the single task and sharing information from the others tasks.
- The complexity and optimal terms depend on the setting hypothesis family $\mathcal{H}$. Given a fixed hypothesis family such as neural network, the complexity is constant. As for the optimal expected loss, throughout this paper we assume $\lambda_{t,i}$ is *much smaller* than the empirical term, which indicates that the hypothesis family $\mathcal{H}$ can learn the multiple tasks with a small expected risk. This is a natural setting in the MTL problem since we want the predefined hypothesis family to learn well for all of the tasks. While a high expected risk means such a hypothesis set cannot perform well, which contradicts our assumption.

In Theorem 1, we have derived a bound based on the $\mathcal{H}$ divergence and applied in the classification problem. Then we proposed another bound based on the Wasserstein distance, which can be applied in the classification and regression problem.

Theorem 2. *Let $\mathcal{H}$ be a hypothesis family from $\mathcal{X}$ to $[0, 1]$, with pseudo-dimension d and each member $h \in \mathcal{H}$ is K Lipschitz. If we have T tasks generated by the underlying distribution and labelling function $\{(\mathcal{D}_1, f_1), \ldots, (\mathcal{D}_T, f_T)\}$ with observation numbers $m_1, \ldots, m_T$. If we adopt Wasserstein-1 distance as a similarity metric with cost function $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$, then for any simplex $\boldsymbol{\alpha}_t \in \Delta^T$, and for $\delta \in (0, 1)$, with a probability at least $1 - \delta$, for $h_1, \ldots, h_T \in \mathcal{H}$, we have:*

$$\frac{1}{T}\sum_{t=1}^{T} R_t(h_t) \leq \underbrace{\frac{1}{T}\sum_{t=1}^{T} \hat{R}_{\boldsymbol{\alpha}_t}(h_t)}_{\text{Weighted empirical loss}} + \underbrace{C_1 \sum_{t=1}^{T}\Big(\sqrt{\sum_{j=1}^{T}\frac{\boldsymbol{\alpha}_{t,j}^2}{\beta_j}}\Big)}_{\text{Coefficient regularization}}$$
$$+ \underbrace{\frac{2K}{T}\sum_{t=1}^{T}\sum_{i=1}^{T}\boldsymbol{\alpha}_{t,i} W_1(\hat{D}_t, \hat{D}_i)}_{\text{Empirical distribution distance}} + \underbrace{C_2 + \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{T}\boldsymbol{\alpha}_{t,i}\lambda_{t,i}}_{\text{Complexity \& optimal expected loss}},$$

where $\beta_i = \frac{m_i}{m}$, $C_1 = 2\sqrt{\frac{2(d\log(\frac{2em}{d}) + \log(\frac{16T}{\delta}))}{m}}$, $C_2 = \frac{2K}{T}\sum_{t=1}^{T}\sum_{i=1}^{T}\gamma_{t,i}$ with $\gamma_{t,i} = \mu_t m_t^{-1/s} + \mu_i m_i^{-1/s} + \sqrt{\log(\frac{2T}{\delta})}(\sqrt{\frac{1}{m_t}} + \sqrt{\frac{1}{m_i}})$ and s and $\mu_\cdot$ are some specified constants.

The proof w.r.t. the Wasserstein-1 distance is analogous to the proof in the $\mathcal{H}$-divergence but with different assumptions. The upper bound of the generalization error shows some intuitions that we should not only minimize the weighted empirical loss, but also minimize the empirical distribution divergence between each task. Moreover, in MTL approaches based on neural networks, these conclusions proposed a theoretical support for understanding the role of *adversarial losses*, which exactly minimize the distribution divergence.

6 Adversarial Multitask Neural Network

From the generalization error upper bound in the MTL framework, we developed a new training algorithm for the Adversarial Multitask Neural Network (AMTNN). It consists in multiple training steps by iteratively optimizing the parameters in the neural network for a given fixed relation coefficient $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T$ and estimating the relation coefficient, given fixed neural network weights.

Moreover, we have three types of parameters in AMTNN: $\boldsymbol{\theta}^f$, $\boldsymbol{\theta}_\cdot^d$ and $\boldsymbol{\theta}_\cdot^h$, corresponding to the parameter for feature

[1] https://arxiv.org/abs/1903.09109

extractor, adversarial loss (distribution similarity) and task loss, respectively.

To simplify the problem, we assume that each task has the same number of observations, i.e., $\beta_i = \frac{1}{T}$, and that regularization will use the l_2 norm of $\boldsymbol{\alpha}_t$.

6.1 Neural Network Parameters Update

Given a fixed $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T$, according to the theoretical bound, we want to minimize the weighted empirical error $\frac{1}{T}\sum_{t=1}^{T} \hat{R}_{\boldsymbol{\alpha}_t}(\boldsymbol{\theta}^f, \boldsymbol{\theta}_t^h)$ and the empirical distribution "distance" $\hat{d}(\mathcal{D}_t, \mathcal{D}_i)$ with $t, i = 1, \ldots, T$. Inspired by [Ganin *et al.*, 2016], the minimization of the distribution "distance" is equivalent to the maximization of the adversarial loss $\hat{E}_{t,i}(\boldsymbol{\theta}^f, \boldsymbol{\theta}_{t,i}^d)$. Overall, we have the following loss function with a trade-off coefficient ρ:

$$\min_{\boldsymbol{\theta}^f, \boldsymbol{\theta}_1^h, \ldots, \boldsymbol{\theta}_t^h} \max_{\boldsymbol{\theta}_{t,i}^d} \sum_{t=1}^{T} \hat{R}_{\boldsymbol{\alpha}_t}(\boldsymbol{\theta}^f, \boldsymbol{\theta}_t^h) + \rho \sum_{i,t=1}^{T} \boldsymbol{\alpha}_{t,i} \hat{E}_{t,i}(\boldsymbol{\theta}^f, \boldsymbol{\theta}_{t,i}^d). \tag{1}$$

It should be noted that for a given task t, the sum loss can be expressed as $\frac{1}{T}\sum_{i=1}^{T} \boldsymbol{\alpha}_{t,i} \sum_{\mathbf{x} \in \hat{D}_i} \ell((\mathbf{x}, y), \boldsymbol{\theta}^f, \boldsymbol{\theta}_t^h)$, with ℓ being the cross entropy loss. This means that the empirical loss is a weighted sum of all of the task losses, determined by task relation coefficient $\boldsymbol{\alpha}_t$. This is coherent with [Murugesan *et al.*, 2016], which does not provide theoretical explanations.

Also, the adversarial loss $\hat{E}_{t,i}(\boldsymbol{\theta}^f, \boldsymbol{\theta}_{t,i}^d)$ is a symmetric metric for which we need to compute $\hat{E}_{t,i}$ only for $t < i$. Motivated by [Ganin *et al.*, 2016], the neural network will output for a pair of observed *unlabeled* tasks $(\hat{\mathcal{D}}_t, \hat{\mathcal{D}}_i)$ a score in $[0, 1]$ to predict from which distribution it comes. Supposing the output function is $g_{t,i}(\mathbf{x}, (\boldsymbol{\theta}^f, \boldsymbol{\theta}_{t,i}^d)) \equiv g_{t,i}(\mathbf{x})$, the adversarial loss will be the following under different distance metrics:

$\mathcal{H}$ divergence:

$$\hat{E}_{t,i} = \sum_{\mathbf{x} \in \hat{\mathcal{D}}_t} \log(g_{t,i}(\mathbf{x})) + \sum_{\mathbf{x} \in \hat{D}_i} \log(1 - g_{t,i}(\mathbf{x}));$$

Wasserstein-1 distance: Since the primal form has a high computational complexity, we adopted the same strategy as [Arjovsky *et al.*, 2017] by estimating the empirical Kantorovich-Rubinstein duality of Wasserstein-1 distance, which leads to $W_1(\hat{D}_t, \hat{D}_i) = \frac{1}{K} \sup_{\|f\| \le K} \left(\mathbb{E}_{\mathbf{x} \in \hat{\mathcal{D}}_t}[g_{t,i}(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \in \hat{\mathcal{D}}_i}[g_{t,i}(\mathbf{x})] \right)$. Combining it with the result of Theorem 2, we can derive that

$$\hat{E}_{t,i} = \mathbb{E}_{\mathbf{x} \in \hat{\mathcal{D}}_t}[g_{t,i}(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \in \hat{\mathcal{D}}_i}[g_{t,i}(\mathbf{x})].$$

6.2 Relation Coefficient Updating

The second step after updating the neural network parameter, we need to re-estimate the coefficients $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T$ when giving fixed $\boldsymbol{\theta}^f, \boldsymbol{\theta}_\cdot^h, \boldsymbol{\theta}_\cdot^d$. According to the theoretical guarantees, we need to solve the following convex constraint optimization problem:

$$\min_{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T} \sum_{t=1}^{T} \hat{R}_{\boldsymbol{\alpha}_t}(\boldsymbol{\theta}^f, \boldsymbol{\theta}_t^h) + \kappa_1 \sum_{t=1}^{T} \|\boldsymbol{\alpha}_t\|_2 + \kappa_2 \sum_{i,t=1}^{T} \boldsymbol{\alpha}_{t,i} \hat{d}_{t,i}(\boldsymbol{\theta}^f, \boldsymbol{\theta}_{t,i}^d), \tag{2}$$
$$\text{s.t.} \quad \|\boldsymbol{\alpha}_t\|_1 = 1, \quad \boldsymbol{\alpha}_{t,i} \ge 0 \quad \forall t, i,$$

where κ_1 and κ_2 are hyper-parameters and $\hat{d}_\cdot$ is the estimated distribution "distance". This distribution "distance" may have different forms with according to the similarity metric used:

$\mathcal{H}$ divergence: According to [Pentina and Lampert, 2017; Ben-David *et al.*, 2010; Ganin *et al.*, 2016], the distribution "distance" is proportional to the accuracy of the

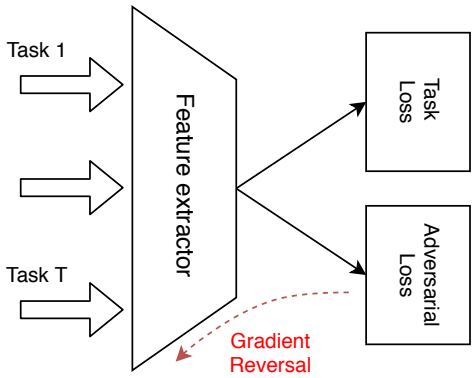


Figure 1: General framework of Adversarial Multitask Neural Network (AMTNN).

Algorithm 1 AMTNN updating algorithm

Require: Samples from different tasks $\{\hat{\mathcal{D}}_t\}_{t=1}^T$, initial coefficients $\{\boldsymbol{\alpha}_t\}_{t=1}^T$ and learning rate η

Ensure: Neural network parameters $\boldsymbol{\theta}^f$, $\boldsymbol{\theta}_\cdot^h$, $\boldsymbol{\theta}_\cdot^d$ and relationship coefficient $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T$

1: **for** mini-batch of samples $\{(\mathbf{x}_t^b, \mathbf{y}_t^b)\}$ from $\{\hat{\mathcal{D}}_t\}_{t=1}^T$ **do**

2: For each the distribution pair (t, i) with $t < i$, compute the adversarial loss $\hat{E}_{t,i}(\boldsymbol{\theta}^f, \boldsymbol{\theta}_{t,i}^d)$.

3: For each task t, define the empirical loss matrix $\hat{R}_{t,i} = \sum_{(\mathbf{x}_i^b, \mathbf{y}_i^b) \in \hat{D}_i} \ell((\mathbf{x}_i^b, \mathbf{y}_i^b), \boldsymbol{\theta}^f, \boldsymbol{\theta}_t^h)$ and compute the label loss:

$$\hat{R}_{\boldsymbol{\alpha}_t} = \sum_{i=1}^{T} \boldsymbol{\alpha}_{t,i} \hat{R}_{t,i}$$

4: Update $\boldsymbol{\theta}^f, \boldsymbol{\theta}_t^h$: $\boldsymbol{\theta}_t^h = \boldsymbol{\theta}_t^h - \eta \frac{\partial \hat{R}_{\boldsymbol{\alpha}_t}}{\partial \boldsymbol{\theta}_t^h}$ and $\boldsymbol{\theta}^f = \boldsymbol{\theta}^f - \eta \left(\sum_{t=1}^{T} \frac{\partial \hat{R}_{\boldsymbol{\alpha}_t}}{\partial \boldsymbol{\theta}^f} + \sum_{t,i:t<i}^{T} (\boldsymbol{\alpha}_{t,i} + \boldsymbol{\alpha}_{i,t}) \frac{\partial \hat{E}_{t,i}}{\partial \boldsymbol{\theta}^f} \right)$

5: Update $\boldsymbol{\theta}_{t,i}^d$ $(t < i)$: $\boldsymbol{\theta}_{t,i}^d = \boldsymbol{\theta}_{t,i}^d + \eta \Big((\boldsymbol{\alpha}_{t,i} + \boldsymbol{\alpha}_{i,t}) \frac{\partial \hat{E}_{t,i}}{\partial \boldsymbol{\theta}_{t,i}^d} \Big)$

6: **end for**

7: Re-estimate $\{\boldsymbol{\alpha}_t\}_{t=1}^T$ by optimizing over Eq. (2).

discriminator $\boldsymbol{\theta}^d_{\cdot}$, i.e., we applied $g_{t,i}(\mathbf{x})$ to predict $\mathbf{x}$ coming from distribution t or i. The prediction accuracy reflects the difficulty to distinguish two distributions. Hence, we set $\hat{d}_{t,i}$ as the accuracy of the discriminator $g_{t,i}(\mathbf{x})$;

Wasserstein-1 distance: According to [Arjovsky *et al.*, 2017], the approximation $\hat{d}_{t,i} = -\hat{E}_{t,i}$ is used.

We also assume $\hat{d}_{t,t} = 0$ since the discriminator cannot distinct from two identical distributions. Moreover, the expected loss $\boldsymbol{\alpha}_t \lambda_{t,\cdot}$ is omitted since we assume that $\lambda_{t,\cdot}$ is much smaller than the empirical term. Then, we only use the empirical parts to re-estimate the relationship coefficient.

As it is mentioned in the theoretical part, the L_2 norm regularization aims at preventing all of the relation coefficient from being concentrated on the current task $\boldsymbol{\alpha}_{t,t}$. The theoretical bound proposed an elegant interpretation for training AMTNN, which is shown in Algorithm 1.

6.3 Training Algorithm

The general framework of the neural network is shown in Fig. 1. We propose a complete iteration step on how to update the neural network parameters and relation coefficients in Algorithm 1. When updating the feature extraction parameter $\boldsymbol{\theta}^f$, we applied *gradient reversal* [Ganin *et al.*, 2016] in the training procedure. We also add the *gradient penalty* [Gulrajani *et al.*, 2017] to improve the Lipschitz property when training with the adversarial loss based on Wasserstein distance.

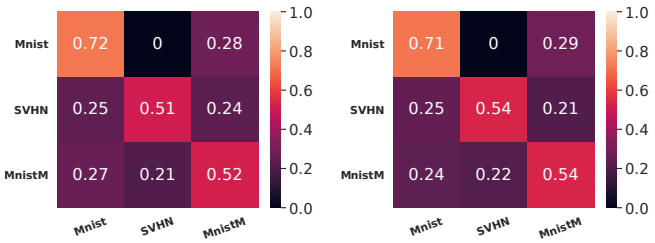
7 Experiments

We evaluate the modified AMTNN method on two benchmarks, that is the digits datasets and the Amazon sentiment dataset. We also consider the following approaches, as baselines to make comparisons: (1) MTL_uni: the vanilla MTL framework where $\frac{1}{T}\sum_{t=1}^{T} \hat{R}_t(\boldsymbol{\theta}^f, \boldsymbol{\theta}^h_t)$ is minimized; (2) MTL_weighted: minimizing $\frac{1}{T}\sum_{t=1}^{T} \hat{R}_{\boldsymbol{\alpha}_t}(\boldsymbol{\theta}^f, \boldsymbol{\theta}^h_t)$, computation of $\boldsymbol{\alpha}_t$ depending on $\hat{R}_{t,i}$, similarly to [Murugesan *et al.*, 2016]; (3) MTL_disH and MTL_disW: we apply the same type of loss function but with two different adversarial losses ($\mathcal{H}$ divergence and Wasserstein distance) and a general neural network without a special part for the NLP [Liu *et al.*, 2017]; (4) AMTNN_H and AMTNN_W: proposed approaches with two different adversarial losses, $\mathcal{H}$ divergence and Wasserstein distance respectively.

7.1 Digit Recognition

We first evaluate our algorithm on three benchmark datasets of digit recognition, which are datasets, MNIST, MNIST_M, and SVHN. The MTL setting is to jointly allow a system to learn to recognize the digits from the three datasets, which can differ significantly. In order to show the effectiveness of MTL, only a small portion of the original dataset is used for training (i.e., 3K, 5K and 8K for each task).

We use the LeNet-5 architecture and define the feature extractor $\boldsymbol{\theta}^f$ as the two convolutional layers of the network, followed by multiple blocks of two fully connected layers as label prediction parameter $\boldsymbol{\theta}^h_{\cdot}$ and discriminator parameter $\boldsymbol{\theta}^d_{\cdot}$. Five repetitions are conducted for each approach, and the average test accuracy (%) is reported in Table 1. We also show the estimated coefficient $\{\boldsymbol{\alpha}_t\}_{t=1}^3$ of AMTNN_H and AMTNN_W, in Fig. 2.



(a) AMTNN_W (b) AMTNN_H

Figure 2: Estimated task relation coefficients matrix from the two proposed algorithms, with training set of 8K instances.

Discussions

Reported results show that the proposed approaches outperform all of the baselines in the task average and also in most single tasks. Particularly for the AMTNN_W, it outperforms the baselines with $1.0\% \sim 2.9\%$ in the test accuracy. The reason can be that the Wasserstein-1 distance is more efficient for measuring the high dimensional distribution, which has been verified theoretically [Redko *et al.*, 2017]. Moreover, the $\mathcal{H}$ divergence-based approach (AMTNN_H) outperforms the baselines without significant increment ($< 0.3\%$). The reason may be that the VC-dimension with $\mathcal{H}$ divergence is not a good metric for measuring a high dimensional complex dataset, coherently with [Li *et al.*, 2018].

As for the coefficients $\boldsymbol{\alpha}_t$, the proposed algorithm appears robust at estimating these task relationships, with almost identical values under different similarity metrics. Moreover, in contrast to the previous approaches, we obtain a non-symmetric matrix with a better interpretability. For instance, when learning for the MNIST dataset, only information from MNIST_M is used, which is reasonable since these two tasks have the same digits configurations with different background, while SVHN is different in most ways (i.e., digits taken from street view house numbers). However, when learning MNIST_M, information from SVHN is beneficial because it provides some information on the background, which is absent from MNIST but similar to MNIST_M. Therefore, the information of both tasks are involved in training for MNIST_M.

In order to show the role of the weighted sum, we use t-SNE to visualise in Fig. 3 the embedded space of the MNIST task from the training data. Information from SVHN is not relevant for learning MNIST as $\boldsymbol{\alpha}_{1,2} = 0$ (see Fig. 2), such that SVHN data is arbitrarily distributed in the embedded space without influence on the final result. At the same time, information from MNIST_M is used for training on the MNIST task ($\boldsymbol{\alpha}_{1,3} = 0.28$), which can be seen by a slight overlap in the embedded space. From that perspective, the role of weighted loss, which helps us to achieve some reasonable modifications of the decision boundary, is trained by the relevant and current tasks jointly. For a small scale task (typically the MTL

Approach	3K				5K				8K			
	MNIST	**MNIST_M**	**SVHN**	Average	**MNIST**	**MNIST_M**	**SVHN**	Average	**MNIST**	**MNIST_M**	**SVHN**	Average
MTL_uni	93.23	76.85	57.20	75.76	97.41	77.72	67.86	81.00	97.73	83.05	71.19	83.99
MTL_weighted	89.09	73.69	68.63	77.13	91.43	74.07	73.81	79.77	92.01	76.69	73.77	80.82
MTL_disH	89.91	**81.13**	70.31	80.45	91.92	82.68	73.27	82.62	92.96	**85.04**	78.50	85.50
MTL_disW	96.77	80.38	68.40	81.85	95.47	**83.48**	72.66	83.87	98.09	84.13	74.37	85.53
AMTNN_H	**97.47**	77.87	71.26	82.20	**97.94**	76.28	76.06	83.43	**98.28**	82.75	76.63	85.89
AMTNN_W	97.20	80.70	**76.93**	**84.95**	97.67	82.50	**76.36**	**85.51**	98.01	82.53	**79.97**	**86.84**

Table 1: Average test accuracy (in %) of MTL algorithms on the digits datasets.

Approach	1000 examples					1600 examples				
	Book	**DVDs**	**Kitchen**	**Elec**	Average	**Book**	**DVDs**	**Kitchen**	**Elec**	Average
MTL_uni	81.31	78.44	87.07	84.57	82.85	81.35	80.14	86.54	87.50	83.88
MTL_weighted	81.88	79.02	86.91	85.31	83.28	80.72	81.20	87.60	88.12	84.41
MTL_disH	81.23	78.12	87.34	84.82	82.88	**81.92**	79.86	87.79	87.31	84.22
MTL_disW	81.13	78.38	87.11	84.82	82.86	81.88	79.81	87.07	87.69	84.11
AMTNN_H	**82.36**	79.24	**87.42**	85.53	**83.64**	80.82	**81.54**	**88.27**	88.17	**84.70**
AMTNN_W	81.68	**79.38**	87.27	**85.66**	83.50	81.20	80.38	87.69	**88.46**	84.44

Table 2: Average test accuracy (in %) of MTL algorithms in the sentiment dataset.

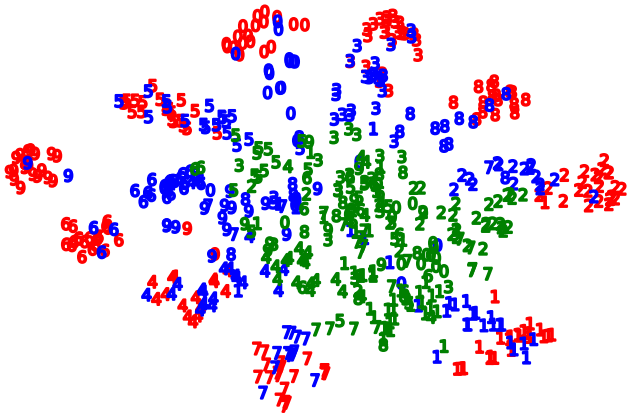


Figure 3: t-SNE in the feature space of task MNIST in AMTNN_W for 8K. Red: MNIST dataset; blue: MNIST_M dataset; green: SVHN data set.

scenario), during the test procedure, the agent predicts the labels by borrowing its neighbors (relevant tasks) information. This is coherent with the Probabilistic Lipschitzness condition [Urner and Ben-David, 2013].

7.2 Sentiment Analysis

We also evaluate the proposed algorithm on *Amazon reviews* datasets. We extract reviews from four product categories: Books, DVD, Electronics and Kitchen appliances. Reviews datasets are pre-processed with the same strategy proposed by [Ganin *et al.*, 2016]: 10K dimensional input features of uni-gram/bi-gram occurrences and binary output labels $\{0, 1\}$, Label 0 is given if the product is ranked less than 3 stars, otherwise label 1 is given for products above 3 stars. Results are reported for two sizes of labelled training sets, that is 1000 and 1600 examples in each product category. The output of the first fully connected layers as feature extractor parameters $\boldsymbol{\theta}^f$ and several sets of two fully-connected layers are given as discriminator $\boldsymbol{\theta}^d_\cdot$ and label predictor $\boldsymbol{\theta}^h_\cdot$, with test accuracy (%) reported in Table 2 as an average over 5 repetitions.

Discussions

We found the proposed approaches outperform all of the baselines in the task average and also in the most tasks. Meanwhile, we observed that the role of adversarial loss (MTL_disH, MTL_disW, AMTNN_H and AMTNN_W) is not that significant (gains $< 0.25\%$), compared to the results on the digits datasets. The possible reason is that we applied the algorithm on the pre-processed feature instead of the original feature, making the discriminator $\boldsymbol{\theta}^d_\cdot$ less powerful in the feature adaptation. By the contrary, adding the weighted loss can improve performance by $0.4\% \sim 0.9\%$, enhancing the importance of the role of the explicit similarity, which is also coherent with [Murugesan *et al.*, 2016].

8 Conclusion

In this paper, we propose a principle approach for using the task similarity information in the MTL framework. We first derive an upper bound of the generalization error in the MTL. Then, according to the theoretical results, we design a new training algorithm on the Adversarial Multi-task Neural Network (AMTNN). Finally, the empirical results on the benchmarks are showing that the proposed algorithm outperforms the baselines, reaffirming the benefits of theoretical insight in the algorithm design. In the future, we want to extend to a more general transfer learning scenario such as the different outputs space.

Acknowledgments

This work was made possible with funding from NSERC-Canada, Mitacs, Prompt-Québec, E Machine Learning and Thales Canada. We thank Annette Schwerdtfeger and Fan Zhou for proofreading this manuscript.

References

[Arjovsky *et al.*, 2017] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[Ben-David *et al.*, 2010] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

[Ganin *et al.*, 2016] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[Janati *et al.*, 2018] Hicham Janati, Marco Cuturi, and Alexandre Gramfort. Wasserstein regularization for sparse multi-task regression. *arXiv preprint arXiv:1805.07833*, 2018.

[Kendall *et al.*, 2018] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[Konstantinov and Lampert, 2019] Nikola Konstantinov and Christoph Lampert. Robust learning from untrusted sources. *arXiv preprint arXiv:1901.10310*, 2019.

[Kremer *et al.*, 2018] Jan Kremer, Lasse Borgholt, and Lars Maaløe. On the inductive bias of word-character-level multi-task learning for speech recognition. *arXiv preprint arXiv:1812.02308*, 2018.

[Lee and Raginsky, 2018] Jaeho Lee and Maxim Raginsky. Minimax statistical learning with wasserstein distances. In *Advances in Neural Information Processing Systems*, pages 2692–2701, 2018.

[Li *et al.*, 2018] Yitong Li, David E Carlson, et al. Extracting relationships by multi-domain matching. In *Advances in Neural Information Processing Systems*, pages 6799–6810, 2018.

[Liu *et al.*, 2017] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-task learning for text classification. *arXiv preprint arXiv:1704.05742*, 2017.

[Long *et al.*, 2017] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and S Yu Philip. Learning multiple tasks with multilinear relationship networks. In *Advances in Neural Information Processing Systems*, pages 1594–1603, 2017.

[Maurer *et al.*, 2016] Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *The Journal of Machine Learning Research*, 17(1):2853–2884, 2016.

[Murugesan and Carbonell, 2017] Keerthiram Murugesan and Jaime Carbonell. Active learning from peers. In *Advances in Neural Information Processing Systems*, pages 7008–7017, 2017.

[Murugesan *et al.*, 2016] Keerthiram Murugesan, Hanxiao Liu, Jaime Carbonell, and Yiming Yang. Adaptive smoothed online multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4296–4304, 2016.

[Pentina and Lampert, 2017] Anastasia Pentina and Christoph H Lampert. Multi-task learning with labeled and unlabeled tasks. In *International Conference on Machine Learning*, pages 2807–2816, 2017.

[Rakotomamonjy *et al.*, 2018] Alain Rakotomamonjy, Abraham Traore, Maxime Berar, Rémi Flamary, and Nicolas Courty. Wasserstein distance measure machines. *arXiv preprint arXiv:1803.00250*, 2018.

[Redko *et al.*, 2017] Ievgen Redko, Amaury Habrard, and Marc Sebban. Theoretical analysis of domain adaptation with optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 737–753. Springer, 2017.

[Ruder, 2017] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[Urner and Ben-David, 2013] Ruth Urner and Shai Ben-David. Probabilistic lipschitzness a niceness assumption for deterministic labels. In *In Learning Faster from Easy Data - Workshop @ NIPS*, 2013.

[Wang and Pineau, 2015] Boyu Wang and Joelle Pineau. Online boosting algorithms for anytime transfer and multitask learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[Zhang and Yang, 2017] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[Zhang and Yeung, 2010] Yu Zhang and Dit Yan Yeung. A convex formulation for learning task relationships in multi-task learning. In *26th Conference on Uncertainty in Artificial Intelligence, UAI 2010, Catalina Island, CA, United States, 8-11 July 2010, Code 86680*, 2010.

[Zhang, 2015] Yu Zhang. Multi-task learning and algorithmic stability. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.