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]

⋆ changjian.shui.1@ulaval.ca; [1] Université Laval, [2] University of Pennsylvania

Introduction

- MTL: Learning a set of related tasks by exploiting the shared knowledge
- Intuition: *Tasks that are alike should be treated alike*
- *Explicit* similarity: minimize a weighted sum of loss, similar tasks are assigned higher weights [1]
- *Implicit* similarity: minimizing the distribution divergence between the tasks by constructing shared features [2]

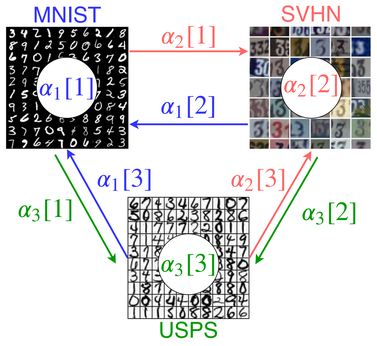
Contributions

Understanding the **fundamental implications** of incorporating task similarity information in MTL algorithms:

- *Why* should we combine explicit and implicit similarity knowledge in the MTL framework
- *How* can we use it for the practice

Problem Setup

- Find T hypothesis $\{h_t\}_{t=1}^T$ from observed tasks $\{\hat{\mathcal{D}}_t := (x_i, y_i)_{i=1}^m\}_{t=1}^T$
- Generalization error: $\frac{1}{T}\sum_{t=1}^T R_t(h_t)$ with $R_t(h_t) = \mathbb{E}_{(x,y)\sim\mathcal{D}_t}\ell(h_t(\mathbf{x}), y)$
- *Relation coefficients*: $\{\boldsymbol{\alpha}_t\}_{t=1}^T$, each $\boldsymbol{\alpha}_t$ is T simplex
- Empirical weighted loss for each task t: $\hat{R}_{\boldsymbol{\alpha}_t}(h) = \sum_{i=1}^T \boldsymbol{\alpha}_t[i]\hat{R}_i(h)$ with $\hat{R}_i(h) = \frac{1}{m}\sum_{(x,y)\sim\hat{D}_i}\ell(h(x), y)$
- Metrics for measuring task similarity: $\mathcal{H}$ divergence and Wasserstein-1 distance



Theoretical Results

Theorem 1 (Informal) *Supposing the transport cost in the Wasserstein distance is* $c(\mathbf{x}, \mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|_2$, *with high probability* $\geq 1-\delta$, *we have:*

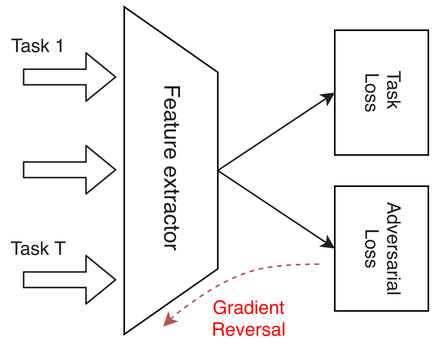
$$\frac{1}{T}\sum_{t=1}^T R_t(h_t) \leq \underbrace{\frac{1}{T}\sum_{t=1}^T \hat{R}_{\boldsymbol{\alpha}_t}(h_t)}_{\text{Weighted empirical loss}} + \underbrace{C_1\sum_{t=1}^T \|\boldsymbol{\alpha}_t\|_2}_{\text{Coefficient regularization}}$$

$$+ \underbrace{\frac{2K}{T}\sum_{t=1}^T\sum_{i=1}^T \boldsymbol{\alpha}_t[i]W_1(\hat{D}_t, \hat{D}_i)}_{\text{Empirical distribution distance}} + \underbrace{C_2 + \frac{1}{T}\sum_{t=1}^T\sum_{i=1}^T \boldsymbol{\alpha}_t[i]\lambda_{t,i}}_{\text{Complexity and optimal expected loss}}$$

C_1 *and* C_2 *constants related with Lipschtiz constant* K, *pseudo-dim* d, m, T *and* δ.

Training Algorithm

Using the theoretical results in training adversarial multitask neural network (AMTNN)



- Neural networks parameters $\boldsymbol{\theta}^f$; $\boldsymbol{\theta}^h_\cdot$; $\boldsymbol{\theta}^d_\cdot$
- Relation coefficients $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T$
- Alternative optimization over two kinds of parameters at each training epoch:

Step 1: Neural net parameter updating

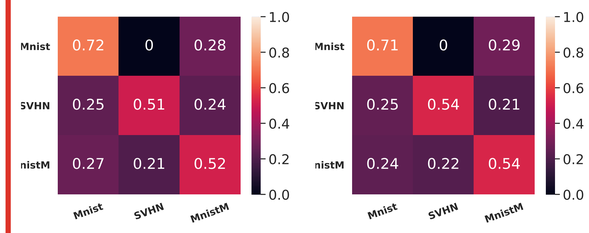
$$\min_{\boldsymbol{\theta}^f, \boldsymbol{\theta}^h_1, \ldots, \boldsymbol{\theta}^h_t} \max_{\boldsymbol{\theta}^d_{t,i}} \sum_{t=1}^T \hat{R}_{\boldsymbol{\alpha}_t}(\boldsymbol{\theta}^f, \boldsymbol{\theta}^h_t) + \rho \sum_{i,t=1}^T \boldsymbol{\alpha}_t[i]\hat{E}_{t,i}(\boldsymbol{\theta}^f, \boldsymbol{\theta}^d_{t,i})$$

Step 2: Relation coefficient updating

$$\min_{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_T} \sum_{t=1}^T \hat{R}_{\boldsymbol{\alpha}_t}(\boldsymbol{\theta}^f, \boldsymbol{\theta}^h_t) + \kappa_1 \sum_{t=1}^T \|\boldsymbol{\alpha}_t\|_2 + \kappa_2 \sum_{i,t=1}^T \boldsymbol{\alpha}_t[i]\hat{d}_{t,i}(\boldsymbol{\theta}^f, \boldsymbol{\theta}^d_{t,i}),$$

$$\text{s.t.} \quad \|\boldsymbol{\alpha}_t\|_1 = 1, \quad \boldsymbol{\alpha}_t[i] \geq 0 \quad \forall t, i,$$

Robust Relation Coefficient



(a) AMTNN_W (b) AMTNN_H

Figure 1: Estimated task relation coefficients matrix $\{\boldsymbol{\alpha}_t\}_{t=1}^3$ with training set of 8K instances.

- Robust to the similarity metric
- Asymmetric relation coefficients (e.g., task SVHN is not helpful for task MNIST)

Role of Weighted Sum

Similar task naturally extends the decision boundary of the original task.

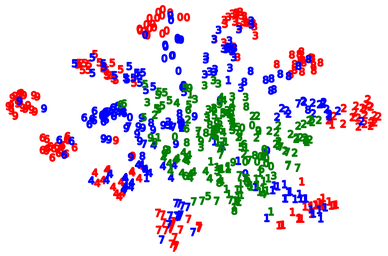


Figure 2: t-SNE in the feature space of task MNIST in AMTNN_W 8K. Red: MNIST; Blue: MNIST_M; Green: SVHN.

Empirical Results

Approach	3K MNIST	3K MNIST_M	3K SVHN	3K Average	5K MNIST	5K MNIST_M	5K SVHN	5K Average	8K MNIST	8K MNIST_M	8K SVHN	8K 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 Book	1000 DVDs	1000 Kitchen	1000 Elec	1000 Average	1600 Book	1600 DVDs	1600 Kitchen	1600 Elec	1600 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.

References

[1] 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.

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

Full paper with details and proofs: `https://arxiv.org/abs/1903.09109`