Semi-Supervised Domain Adaptation with Source Label Adaptation

Yu-Chu Yu Hsuan-Tien Lin

Department of Computer Science & Information Engineering

National Taiwan University

Dec 2nd, 2023

0 Outline

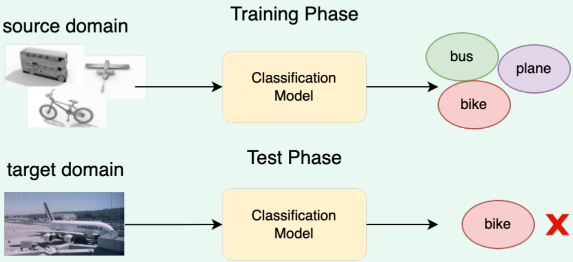
1 Outline

1 Outline

1 Domain Adaptation

Domain Adaptation

Domain Adaptation (DA) is a generalized image classification problem, where we assume that the training and test data are drawn from two different domains.



The two different domains somehow share some **invariant features**.

Semi-Supervised Domain Adaptation

In Semi-Supervised Domain Adaptation (SSDA), few target labels are available.

Usually, we discuss about the **one or three shot** case, where we can only have access to **one or three labels** for each class.

Semi-Supervised Domain Adaptation

In Semi-Supervised Domain Adaptation (SSDA), few target labels are available.

Usually, we discuss about the **one or three shot** case, where we can only have access to **one or three labels** for each class.

Goal

Learning the **invariant features** from both domains, transfering knowledge from a source domain to another target domain.

1 Outline

1 Typical Solution

Typical loss function for SSDA

$$L_{\mathsf{SSDA}} = \underbrace{L^s}_{\text{source loss}} + \underbrace{L^\ell}_{\text{labeled target loss}} + \underbrace{L^u}_{\text{unlabeled target loss}}$$

1 Typical Solution

Typical loss function for SSDA

$$L_{\text{SSDA}} = \underbrace{L^s}_{\text{source loss}} + \underbrace{L^\ell}_{\text{labeled target loss}} + \underbrace{L^u}_{\text{unlabeled target loss}}$$

Baseline solution: S+T

$$L_{\text{SSDA}}(g|S,L) = \underbrace{\frac{1}{|S|}\sum_{i=1}^{|S|} H(g(\mathbf{x}_i^s), \mathbf{y}_i^s)}_{L^s} + \underbrace{\frac{1}{|L|}\sum_{i=1}^{|L|} H(g(\mathbf{x}_i^\ell), \mathbf{y}_i^\ell)}_{L^\ell}$$

1 Typical Solution

Typical loss function for SSDA

$$L_{\text{SSDA}} = \underbrace{L^s}_{\text{source loss}} + \underbrace{L^\ell}_{\text{labeled target loss}} + \underbrace{L^u}_{\text{unlabeled target loss}}$$

Baseline solution: S+T

$$L_{\text{SSDA}}(g|S,L) = \underbrace{\frac{1}{|S|}\sum_{i=1}^{|S|} H(g(\mathbf{x}_i^s), \mathbf{y}_i^s)}_{L^s} + \underbrace{\frac{1}{|L|}\sum_{i=1}^{|L|} H(g(\mathbf{x}_i^\ell), \mathbf{y}_i^\ell)}_{L^\ell}$$

State-of-the-art Solution

Based on S+T, explore the usage of unlabeled data and design fancy L^u.

1 Challenge of S+T I

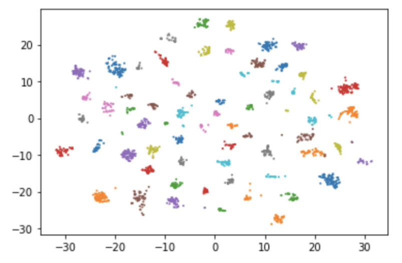
Domain Shift

The feature space derived by S+T is biased. This is a known issue named **domain shift**.

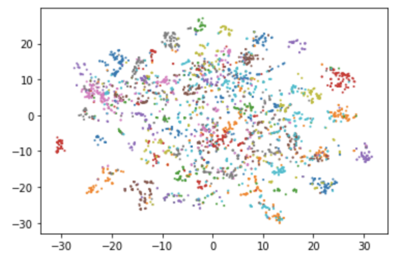
Domain Shift

The feature space derived by S+T is biased. This is a known issue named **domain shift**.

The feature space derived by S+T



Source data



Target data

1 Challenge of S+T III

Misalignment

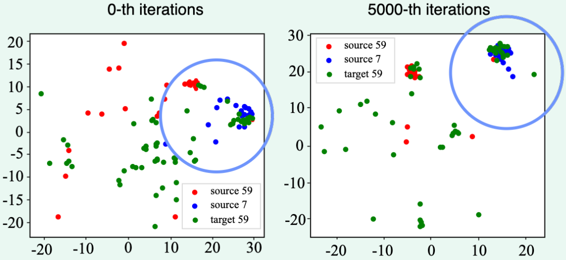
In the training procedure, some target data is misaligned to the wrong classes. For example, the source data in 7th class misguides target data in 59th class to 7th class.

1 Challenge of S+T III

Misalignment

In the training procedure, some target data is misaligned to the wrong classes. For example, the source data in 7th class misguides target data in 59th class to 7th class.

Misalignment of target data



(Partial) Confusion Matrix

- The (partial) confusion matrix shows that, about one-third target data in 59th class are mis-predicted as 7th class.
- Only about 20% data are predicted correctly.

True\Pred	Class 7	Class 59	Class 41	Others
Class 59	33.3%	19.2%	15.2%	32.3%

Source Labels are noisy

Observed by the above case, it seems that the source labels are noisy in the target data point of view.

Source Labels are noisy

Observed by the above case, it seems that the source labels are noisy in the target data point of view.

Recall: S+T

$$L_{\mathsf{SSDA}}(g|S,L) = \underbrace{\frac{1}{|S|}\sum_{i=1}^{|S|} H(g(\mathbf{x}_i^s), \mathbf{y}_i^s)}_{\text{Does it make sense?}} + \frac{1}{|L|}\sum_{i=1}^{|L|} H(g(\mathbf{x}_i^\ell), \mathbf{y}_i^\ell)$$

1 Noisy Source Labels

Source Labels are noisy

Observed by the above case, it seems that the source labels are noisy in the target data point of view.

Recall: S+T

$$L_{\mathsf{SSDA}}(g|S, L) = \underbrace{\frac{1}{|S|}\sum_{i=1}^{|S|} H(g(\mathbf{x}_i^s), \mathbf{y}_i^s)}_{\text{Does it make sense?}} + \frac{1}{|L|}\sum_{i=1}^{|L|} H(g(\mathbf{x}_i^\ell), \mathbf{y}_i^\ell)$$

Question

- Can we approach DA as a Noisy Label Learning problem?
- How to correct source labels to better fit the target feature space?

2 Outline

2 Domain Adaptation as Noisy Label Learning

Recall: Goal for Domain Adaptation

Find an ideal model g^* that can minimize unlabeled target risk.

2 Domain Adaptation as Noisy Label Learning

Recall: Goal for Domain Adaptation

Find an ideal model g^* that can minimize unlabeled target risk.

Domain Adaptation as Noisy Label Learning

$$\underbrace{\mathbf{y}_i^s}_{\text{Noisy}} \xRightarrow{\text{Correction}} \underbrace{g^*(\mathbf{x}_i^s)}_{\text{Clean}}$$

2 Domain Adaptation as Noisy Label Learning

Recall: Goal for Domain Adaptation

Find an ideal model g^* that can minimize unlabeled target risk.

Domain Adaptation as Noisy Label Learning

$$\underset{\text{Noisy}}{\mathbf{y}_i^s} \xRightarrow{\text{Correction}} \underset{\text{Clean}}{g^*(\mathbf{x}_i^s)}$$

Review: Noisy Label Learning

Reed et al. (2014) proposed a way to dynamically correct noisy labels based on self-prediction. We term it: Label Correction with Self-Prediction.

$$\tilde{\mathbf{y}}_i^s = (1-\alpha)\cdot \mathbf{y}_i^s + \alpha \cdot g(\mathbf{x}_i^s) \tag{1}$$

2 Problem in Label Correction with Self-Prediction I

Overfitting Issue

In Domain Adaptation, the model usually overfits to source data, which makes $g(\mathbf{x}_i^s) \approx \mathbf{y}_i^s$.

$$\begin{aligned} \tilde{\mathbf{y}}_i^s &= (1-\alpha) \cdot \mathbf{y}_i^s + \alpha \cdot g(\mathbf{x}_i^s) \\ &\approx (1-\alpha) \cdot \mathbf{y}_i^s + \alpha \cdot \mathbf{y}_i^s = \mathbf{y}_i^s \end{aligned} \tag{2}$$

In this case, doing label correction is nearly equivalent to not doing so.

2 Problem in Label Correction with Self-Prediction II

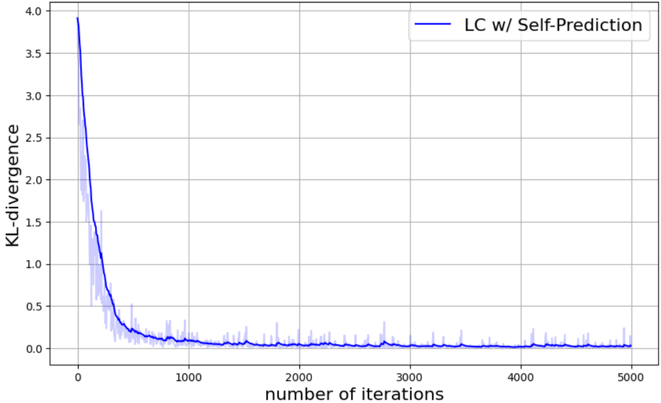


Figure: Average KL divergence from $\mathbf{y}^s$ to $g(\mathbf{x}^s)$ at each iteration. (Office-Home Ar. $\rightarrow$ Cl. with ResNet-34)

2 Challenge of Label Correction with Self-Prediction

Challenge

We need to eliminate supervision from source data.

2 Challenge of Label Correction with Self-Prediction

Challenge

We need to eliminate supervision from source data.

Goal

Find a label adaptation model g_c that can provide view from target data.

$$\tilde{\mathbf{y}}_i^s = (1-\alpha) \cdot \mathbf{y}_i^s + \alpha \cdot g_c(\mathbf{x}_i^s) \tag{3}$$

Challenge

We need to eliminate supervision from source data.

Goal

Find a label adaptation model g_c that can provide view from target data.

$$\tilde{\mathbf{y}}_i^s = (1-\alpha) \cdot \mathbf{y}_i^s + \alpha \cdot g_c(\mathbf{x}_i^s) \tag{3}$$

Few-Shot Learning

- Recall that we have access to few target labels per class. :)
- We can borrow some ideas from few-shot learning.

2 Outline

2 Review: Prototypical Network I

Motivation

Originally designed by Snell et al. (2017) for few-shot learning.

2 Review: Prototypical Network I

Motivation

Originally designed by Snell et al. (2017) for few-shot learning.

Center-based Prototypes

Given a dataset $\{\mathbf{x}_i, y_i\}_{i=1}^N$ and a feature extractor f. The **center** $\mathbf{c}_k$ of k-th class is defined as:

$$\mathbf{c}_k = \frac{1}{N_k} \sum_{i=1}^{N} \mathbb{1}\{y_i = k\} \cdot f(\mathbf{x}_i) \tag{4}$$

where N_k is the number of data in class k.

2 Review: Prototypical Network II

Prototypical Network (Protonet)

Let $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$ collects all centers. $P : X \mapsto Y$ is a prototypical network (protonet) with centers $\mathbf{C}$:

$$P(\mathbf{x})_k = \frac{\exp(-d(f(\mathbf{x}), \mathbf{c}_k))}{\sum_{j=1}^{K} \exp(-d(f(\mathbf{x}), \mathbf{c}_j))} \tag{5}$$

$d : F \times F \mapsto [0, \infty)$ is a distance measure over feature space F.

Remark

Protonet is a **classifier** based on the distance between data and centers.

Protonet with labeled target centers

Given a feature extractor f and labeled target data L, we can:

- Derive labeled target centers $\mathbf{C}_f^\ell$ by eq. 4.
- Build a Protonet with Labeled Target Centers $P_{\mathbf{C}_f^\ell}$ by eq. 5.

Protonet with labeled target centers

Given a feature extractor f and labeled target data L, we can:

- Derive labeled target centers $\mathbf{C}_f^{\ell}$ by eq. 4.
- Build a Protonet with Labeled Target Centers $P_{\mathbf{C}_f^{\ell}}$ by eq. 5.

Ideal target centers

For a protonet, the ideal centers $\mathbf{C}_f^{*}$ should be derived through unlabeled target data $\{(\mathbf{x}_i^u, y_i^u)\}_{i=1}^{|U|}$.

Protonet with labeled target centers

Given a feature extractor f and labeled target data L, we can:

- Derive labeled target centers $\mathbf{C}_f^\ell$ by eq. 4.
- Build a Protonet with Labeled Target Centers $P_{\mathbf{C}_f^\ell}$ by eq. 5.

Ideal target centers

For a protonet, the ideal centers $\mathbf{C}_f^*$ should be derived through unlabeled target data $\{(\mathbf{x}_i^u, y_i^u)\}_{i=1}^{|U|}$.

Challenge

But we have no access to the labels of the unlabeled target data! :(

Pseudo Labeling

With the current model g, the pseudo label $\tilde{y}_i^u$ for an unlabeled data $\mathbf{x}_i^u$ is:

$$\tilde{y}_i^u = \arg\max_k g(\mathbf{x}_i^u)_k \tag{6}$$

Pseudo Labeling

With the current model g, the pseudo label $\tilde{y}_i^u$ for an unlabeled data $\mathbf{x}_i^u$ is:

$$\tilde{y}_i^u = \arg\max_k g(\mathbf{x}_i^u)_k \tag{6}$$

Protonet with Pseudo Centers (PPC)

Let $\tilde{\mathbf{C}}_f = \{\tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_K\}$, where $\tilde{\mathbf{c}}_k$ is the pseudo center derived through $\{\mathbf{x}_i^u, \tilde{y}_i^u\}_{i=1}^{|U|}$ and extractor f. $P_{\tilde{\mathbf{C}}_f}$ is a protonet with pseudo centers.

2 Distance between different centers

Distance between different centers

From / To	labeled target centers	pseudo centers
ideal centers	10.02	4.06

Table: Average L2 Distance from labeled target centers / pseudo centers to ideal centers over the feature space trained by S+T.

Pseudo centers are indeed much closer to the ideal case.

Recall: Source Label Adaptation for SSDA

Find a label adaptation model g_c to correct source labels:

$$\tilde{\mathbf{y}}_i^s = (1 - \alpha) \cdot \mathbf{y}_i^s + \alpha \cdot g_c(\mathbf{x}_i^s)$$

Recall: Source Label Adaptation for SSDA

Find a label adaptation model g_c to correct source labels:

$$\tilde{\mathbf{y}}_i^s = (1-\alpha)\cdot \mathbf{y}_i^s + \alpha \cdot g_c(\mathbf{x}_i^s)$$

PPC as the corrector

Given unlabeled data with pseudo labels $\{(\mathbf{x}_i^u, \tilde{y}_i^u)\}_{i=1}^{|U|}$, and a feature extractor f, we propose to let the protonet with pseudo centers $P_{\tilde{\mathbf{C}}_f}$ be the corrector:

$$\tilde{\mathbf{y}}_i^s = (1-\alpha)\cdot \mathbf{y}_i^s + \alpha \cdot P_{\tilde{\mathbf{C}}_f}(\mathbf{x}_i^s) \tag{7}$$

Recall: Typical loss function for SSDA

$$L_{\mathsf{SSDA}} = \underbrace{L^s + L^\ell}_{\text{standard cross entropy loss}} + L^u$$

2 Framework: Source Label Adaptation for SSDA I

Recall: Typical loss function for SSDA

$$L_{\text{SSDA}} = \underbrace{L^s + L^\ell}_{\text{standard cross entropy loss}} + L^u$$

Label Adaptation Loss

- For each source data $\mathbf{x}_i^s$, compute corrected label $\tilde{\mathbf{y}}_i^s$ by eq. 7.
- We define Label Adaptation Loss $\tilde{L}^s$ as:

$$\tilde{L}^s = \frac{1}{|S|} \sum_{i=1}^{|S|} H(g(\mathbf{x}_i^s), \tilde{\mathbf{y}}_i^s) \quad (8)$$

Framework: Source Label Adaptation for SSDA

$$L_{\text{SSDA w/ SLA}} = \tilde{L}^s + L^\ell + L^u$$

- L^ℓ can be still a standard cross entropy loss for labeled target data.
- L^u can be derived through any SOTA algorithms.

2 Framework: Source Label Adaptation for SSDA II

Framework: Source Label Adaptation for SSDA

$$L_{\text{SSDA w/ SLA}} = \tilde{L}^s + L^\ell + L^u$$

- L^ℓ can be still a standard cross entropy loss for labeled target data.
- L^u can be derived through any SOTA algorithms.

Scalibility

We can easily apply Source Label Adaptation to any SOTA algorithms.

3 Outline

3 Experiment Setups

Datasets

- Office-Home (Venkateswara et al. 2017)
- DomainNet (Peng et al. 2019)

Datasets

- Office-Home (Venkateswara et al. 2017)
- DomainNet (Peng et al. 2019)

Classic State-of-the-art SSDA algorithms

- MME (Saito et al. 2019)
- CDAC (Li et al. 2021)

Datasets

- Office-Home (Venkateswara et al. 2017)
- DomainNet (Peng et al. 2019)

Classic State-of-the-art SSDA algorithms

- MME (Saito et al. 2019)
- CDAC (Li et al. 2021)

Things to Verify

- Can we apply Source Label Adaptation (SLA) to the above methods, and get improvement?

3 Results on Office-Home I

Figure: 3 baseline methods before / after applying SLA on 3-shot Office-Home A -» C case.

3 Results on Office-Home II

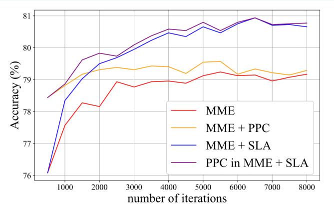
Method	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	Mean
							One-shot						
S+T	50.9	69.8	73.8	56.3	68.1	70.0	57.2	48.3	74.4	66.2	52.1	78.6	63.8
DANN [5]	52.3	67.9	73.9	54.1	66.8	69.2	55.7	51.9	68.4	64.5	53.1	74.8	62.7
ENT [6]	52.9	75.0	76.7	63.2	73.6	73.2	63.0	51.9	79.9	70.4	53.6	81.9	67.9
APE [10]	53.9	76.1	75.2	63.6	69.8	72.3	63.6	58.3	78.6	72.5	60.7	81.6	68.9
DECOTA [31]	42.1	68.5	72.6	60.3	70.4	70.7	60.0	48.8	76.9	71.3	56.0	79.4	64.8
MME [21]	59.6	75.5	77.8	65.7	74.5	74.8	64.7	57.4	79.2	71.2	61.9	82.8	70.4
MME + SLA (ours)	62.1	76.3	78.6	**67.5**	77.1	75.1	66.7	59.9	80.0	**72.9**	64.1	83.8	72.0
CDAC [12]	61.2	75.9	78.5	64.5	75.1	75.3	64.6	59.3	80.0	72.7	61.9	83.1	71.0
CDAC + SLA (ours)	**63.0**	**78.0**	**79.2**	66.9	**77.6**	**77.0**	**67.3**	**61.8**	**80.5**	72.7	**66.1**	**84.6**	**72.9**
							Three-shot						
S+T	54.0	73.1	74.2	57.6	72.3	68.3	63.5	53.8	73.1	67.8	55.7	80.8	66.2
DANN [5]	54.7	68.3	73.8	55.1	67.5	67.1	56.6	51.8	69.2	65.2	57.3	75.5	63.5
ENT [6]	61.3	79.5	79.1	64.7	79.1	76.4	63.9	60.5	79.9	70.2	62.6	85.7	71.9
APE [10]	63.9	81.1	80.2	66.6	79.9	76.8	66.1	65.2	82.0	73.4	66.4	86.2	74.0
DECOTA [31]	64.0	81.8	80.5	68.0	**83.2**	79.0	69.9	68.0	82.1	74.0	**70.4**	**87.7**	75.7
MME [21]	63.6	79.0	79.7	67.2	79.3	76.6	65.5	64.6	80.1	71.3	64.6	85.5	73.1
MME + SLA (ours)	65.9	81.1	80.5	**69.2**	81.9	79.4	69.7	67.4	81.9	**74.7**	68.4	87.4	75.6
CDAC [12]	65.9	80.3	80.6	67.4	81.4	**80.2**	67.5	67.0	81.9	72.2	67.8	85.6	74.8
CDAC + SLA (ours)	**67.3**	**82.6**	**81.4**	**69.2**	82.1	80.1	**70.1**	**69.3**	**82.5**	73.9	70.1	87.1	**76.3**

Table 5. Accuracy (%) on *Office-Home* for 1-shot and 3-shot Semi-Supervised Domain Adaptation (ResNet34).

Question

PPC is an approximation of the ideal model. If PPC has performed well, why not simply use PPC in the model for inference?

Intermediate results in SLA

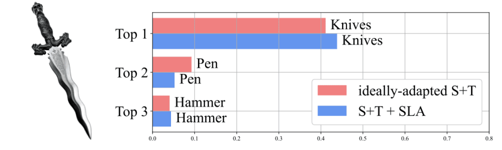


Recall

Recall that we would like to take PPC to approximate the ideal model g^*.

Here we plot the average top-3 probibility of PPC(x^s) and $g^*(x^s)$.

Illustration of the top-3 probability of the adapted labels



Remark

The original source labels is 100% of knives.

4 Outline

Rethinking the usage of source data

- Approach Domain Adaptation as a Noisy Label Learning problem.

4 Conclusion

Rethinking the usage of source data

- Approach Domain Adaptation as a Noisy Label Learning problem.

General Framework

- General framework: Source Label Adaptation for Domain Adaptation
- It can be easily applied to any state-of-the-art algorithm which focuses on the usage of unlabeled data.

4 Conclusion

Rethinking the usage of source data

- Approach Domain Adaptation as a Noisy Label Learning problem.

General Framework

- General framework: Source Label Adaptation for Domain Adaptation
- It can be easily applied to any state-of-the-art algorithm which focuses on the usage of unlabeled data.

Empirical Improvement

- Our method improve 2 representative SOTA algorithms on 2 major datasets for both 1-shot and 3-shot settings.

Thank you for your attention! Any Questions?

Li, J., Li, G., Shi, Y. & Yu, Y. (2021), ‘Cross-domain adaptive clustering for semi-supervised domain adaptation’.
URL: *https://arxiv.org/abs/2104.09415*

Peng, X., Bai, Q., Xia, X., Huang, Z., Saenko, K. & Wang, B. (2019), Moment matching for multi-source domain adaptation, *in* ‘Proceedings of the IEEE International Conference on Computer Vision’, pp. 1406–1415.

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D. & Rabinovich, A. (2014), ‘Training deep neural networks on noisy labels with bootstrapping’.

Saito, K., Kim, D., Sclaroff, S., Darrell, T. & Saenko, K. (2019), ‘Semi-supervised domain adaptation via minimax entropy’.
URL: *https://arxiv.org/abs/1904.06487*

Snell, J., Swersky, K. & Zemel, R. S. (2017), ‘Prototypical networks for few-shot learning’.
URL: *https://arxiv.org/abs/1703.05175*

Venkateswara, H., Eusebio, J., Chakraborty, S. & Panchanathan, S. (2017), Deep hashing network for unsupervised domain adaptation, *in* ‘Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition’, pp. 5018–5027.