AAAI-19

Transferable Curriculum for Weakly-Supervised Domain Adaptation

Yang Shu, Zhangjie Cao, Mingsheng Long*, and Jianmin Wang

School of Software, Tsinghua University, China

KLiss, MOE; BNRist; Research Center for Big Data, Tsinghua University, China

OUTLINE

1. Domain adaption
2. Curriculum Learning
3. Learning with Transferable Curriculum
4. Experiment

Domain adaption

Domain adaptation (Pan and Yang 2010) aims to build learning machines that generalize across **different but relevant domains**.

Source domain

- A different but relevant domains
- Abundant training examples

Target domain

- Target task
- Limited or no training examples

Learning with Transferable Curriculum

this paper addresses the weakly-supervised domain adaptation problem where the **source domain is noisy in labels or features**, and the **target domain is fully unlabeled**.

Goal:

1. Close the domain gap and bound the target risk by learning **transferable features f** and a **robust classifier** across noisy source domain and clean target domain

2. Train a deep network with **a transferable curriculum** to eliminate the negative influence of noisy source samples and enable positive transfer of noiseless source samples

Curriculum Learning

Instead of feeding all samples to the model at once, SPL learns samples from **easy to hard gradually** just like the process of human learning:

Easy images of cat to learn earlier

Complex images of cat to learn later

Learning samples in a Self-Paced way has yielded brilliant results in many applications.
(Khan, Mutlu, and Zhu NIPS'11; Tang, Yang, and Gao MM'12; Basu and Christensen AAAI'13; Zhang et al. IJCAI'16)

Curriculum Learning

Repeat:

1. Find easy examples based on current *curriculum*
2. Use easy data to train the model
3. Update *curriculum* to select harder examples

Until Stopping criterion

$$\min_{\boldsymbol{\theta},\mathbf{w}} E(\boldsymbol{\theta},\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} w_i L\left(\mathbf{y}_i, f\left(\mathbf{x}_i;\boldsymbol{\theta}\right)\right) + R(\mathbf{w};\gamma)$$

Sample loss Curriculum

$\min_{w} wL + R(w;\gamma)$ → The solution of w is inversely proportional to the loss L.

$R(\mathbf{w};\gamma) = -\gamma\|\mathbf{w}\|_1$

$w_i^* = \mathbb{1}\left(\ell_i \leqslant \gamma\right), i = 1,\ldots,n,$

$\min_{f} wL$

Learning with Transferable Curriculum

Step 1: Minimization of distribution shift between the source and target domains

$\boldsymbol{G_f}$: Feature extractor

$\boldsymbol{G_d}$: Discriminator to distinguish the feature representations of the source domain from the target domain

$$E_{G_d} = -\frac{1}{n_s}\sum_{i=1}^{n_s} w\left(\mathbf{x}_i^s\right)\log\left(G_d\left(G_f\left(\mathbf{x}_i^s\right)\right)\right) - \frac{1}{n_t}\sum_{j=1}^{n_t}\log\left(1-G_d\left(G_f\left(\mathbf{x}_j^t\right)\right)\right).$$

ICML'15 **Domain Adversarial Learning**, Ganin and Lempitsky 2015

Learning with Transferable Curriculum

Step 2: Train a classifier for target domain

$\boldsymbol{G_f}$: Feature extractor

$\boldsymbol{G_y}$: Classifier

$$E_{G_y} = \frac{1}{n_s}\sum_{i=1}^{n_s} w(\mathbf{x}_i^s) L_y\left(\mathbf{y}_i^s, G_y\left(G_f\left(\mathbf{x}_i^s\right)\right)\right) + \frac{1}{n_t}\sum_{j=1}^{n_t} H_y\left(G_y\left(G_f\left(\mathbf{x}_j^t\right)\right)\right),$$

Weighted training error in source domain

Entropy loss in target domain

the entropy loss to quantify the uncertainty of a target example's label predictions

$$H_y\left(G_y\left(G_f\left(\mathbf{x}_j^t\right)\right)\right) = -\sum_{c=1}^{m} \hat{y}_{j,c}^t \log \hat{y}_{j,c}^t.$$

Learning with Transferable Curriculum

Step 3: Find easy and transferable examples

simultaneously prioritizes easier *and* transferable examples

$$w(\mathbf{x}_i^s) = \mathbb{1}\left(\ell_i + \lambda\tau_i \leqslant \gamma\right) \text{ where}$$
$$\ell_i = L_y\left(\mathbf{y}_i^s, G_y\left(G_f\left(\mathbf{x}_i^s\right)\right)\right),$$
$$\tau_i = -\log\left(1 - G_d(G_f(\mathbf{x}_i^s))\right)$$

ity $G_d(G_f(\mathbf{x}_i^s))$ indicates the probability of classifying $\mathbf{x}_i^s$ as from the source domain, while $1 - G_d(G_f(\mathbf{x}_i^s))$ indicates the probability of classifying $\mathbf{x}_i^s$ as from the target domain. Thus, $1 - G_d(G_f(\mathbf{x}_i^s))$ is a good indicator to the transferability (similarity) of a source example $\mathbf{x}_i^s$ to the target domain.

Alternating Minimax Optimization

$\boldsymbol{G_f}$: Feature extractor

$\boldsymbol{G_d}$: Discriminator to to distinguish the feature representations of the source domain from the target domain

$\boldsymbol{G_y}$: Classifier

$$(\hat{\boldsymbol{\theta}}_f, \hat{\boldsymbol{\theta}}_y) = \arg\min_{\boldsymbol{\theta}_f, \boldsymbol{\theta}_y} E_{G_y} - E_{G_d},$$

$$(\hat{\boldsymbol{\theta}}_d) = \arg\min_{\boldsymbol{\theta}_d} E_{G_d},$$

$$w(\mathbf{x}_i^s) = \mathbb{1}\left(\ell_i + \lambda\tau_i \leqslant \gamma\right).$$

Experiment

Compared methods:

Baseline

ResNet-50 (He et al. CVPR'16)

Curriculum learning

Self-Paced Learning (**SPL**) (Kumar, Packer, and Koller NIPS'10)
MentorNet (Jiang et al. ICML'18)

Domain adaption methods

Deep Adaptation Network (**DAN**) (Long et al. ICML'15)
Residual Transfer Network (**RTN**) (Long et al. NIPS'16)
Domain Adversarial Neural Network (**DANN**) (Ganin et al. JMLR'16)
Adversarial Discriminative Domain Adaptation (**ADDA**) (Tzeng et al. CVPR'17)

Dataset:

Office-31: 4652 images with 31 classes in 3 distinct domains

Office-Home : consisting of 15,500 images from 65 classes in 4 domains

Bing-Caltech: Bing -> Caltech-256

Source code: https://github.com/thuml

Experiment

Table 1: Classification Accuracy (%) on **Office-31** with 40% Corruption of Labels, Features and Both

Method	Label Corruption							Feature Corruption							Mixed Corruption						
	A→W	W→A	A→D	D→A	W→D	D→W	Avg	A→W	W→A	A→D	D→A	W→D	D→W	Avg	A→W	W→A	A→D	D→A	W→D	D→W	Avg
ResNet	47.2	33.0	47.1	31.0	68.0	58.8	47.5	70.2	55.1	73.0	55.0	94.5	87.2	72.5	58.8	39.1	69.3	37.7	75.2	75.5	59.3
SPL	72.6	50.0	75.3	38.9	83.3	64.6	64.1	75.8	59.7	75.7	56.7	93.9	87.8	74.9	77.3	57.5	78.4	47.5	93.4	83.5	72.9
MentorNet	74.4	54.2	75.0	43.2	85.9	70.6	67.2	76.0	60.3	75.5	59.1	93.4	89.9	75.7	76.8	59.5	78.2	52.3	94.4	89.0	75.0
DAN	63.2	39.0	58.0	36.7	71.6	61.6	55.0	73.9	60.2	72.2	59.6	92.5	88.0	74.4	64.4	45.1	71.2	44.7	79.3	78.3	63.8
RTN	64.6	56.2	76.1	49.0	82.7	71.7	66.7	81.0	**64.6**	81.3	62.3	**95.2**	91.0	79.2	76.7	56.9	**84.1**	56.4	93.0	86.7	75.6
DANN	61.2	46.2	57.4	42.4	74.5	62.0	57.3	71.3	54.1	69.0	54.1	84.5	84.6	69.6	69.7	50.0	69.5	49.1	80.1	79.7	66.4
ADDA	61.5	49.2	61.2	45.5	74.7	65.1	59.5	76.8	62.0	79.8	60.1	93.7	89.3	77.0	69.7	54.5	72.4	56.0	87.5	85.5	70.9
TCL	**82.0**	**65.7**	**83.3**	**60.5**	**90.8**	**77.2**	**76.6**	**84.9**	62.3	**83.7**	**64.0**	93.4	**91.3**	**79.9**	**87.4**	**64.6**	83.1	**62.2**	**99.0**	**92.7**	**81.5**

Table 2: Classification Accuracy (%) on **Office-Home** with 40% Mixed Corruption and **Bing-Caltech** with Native Noises

Method	Office-Home													Bing-Caltech
	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Avg	B→ C
ResNet	27.1	50.7	61.7	41.1	53.8	56.3	40.9	28.0	61.8	51.3	33.0	65.9	47.6	74.4
SPL	32.4	56.0	67.4	41.9	55.3	57.2	47.9	32.9	69.3	60.0	36.2	70.4	52.2	75.3
MentorNet	34.5	57.1	66.7	43.3	56.1	57.6	48.5	34.0	70.2	59.8	37.2	70.4	53.0	75.6
DAN	31.2	52.3	61.2	41.2	53.1	54.6	40.7	30.3	61.5	51.7	36.7	67.4	48.5	75.0
RTN	29.3	57.8	66.3	44.0	**58.6**	58.3	46.0	30.1	67.5	56.3	32.2	69.9	51.4	75.8
DANN	32.9	50.6	60.1	38.6	49.2	50.6	39.9	32.6	60.4	50.5	38.4	67.4	47.6	72.3
ADDA	32.6	52.0	60.6	42.6	53.5	54.3	43.0	31.6	63.1	52.7	37.7	67.5	49.3	74.7
TCL	**38.8**	**62.1**	**69.4**	**46.5**	58.5	**59.8**	**51.3**	**39.9**	**72.3**	**63.4**	**43.5**	**74.0**	**56.6**	**79.0**

Experiment

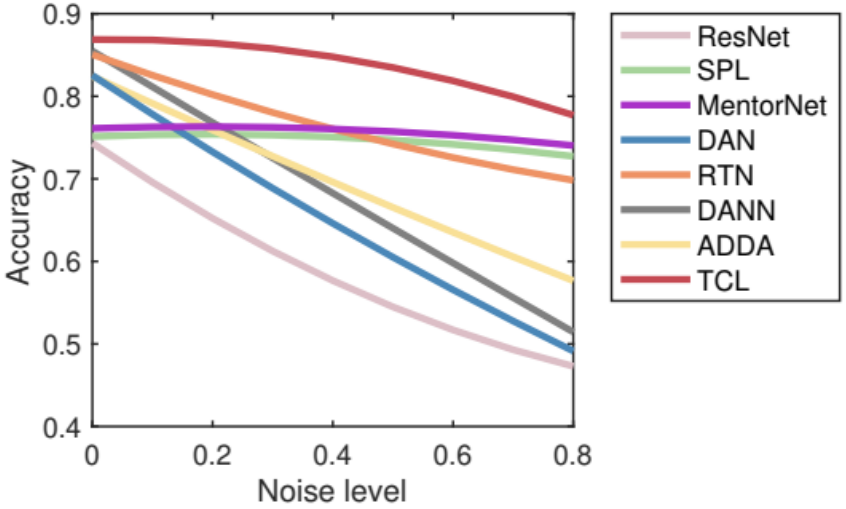
TCL-adversarial_w: removing w for the source data on the domain discriminator;
TCL-classifier_w : removing w for the source data on the label classifier
TCL-easiness: removing easiness term ℓ when updateing w
TCL-transferability: removing transferability term τ when updateing w

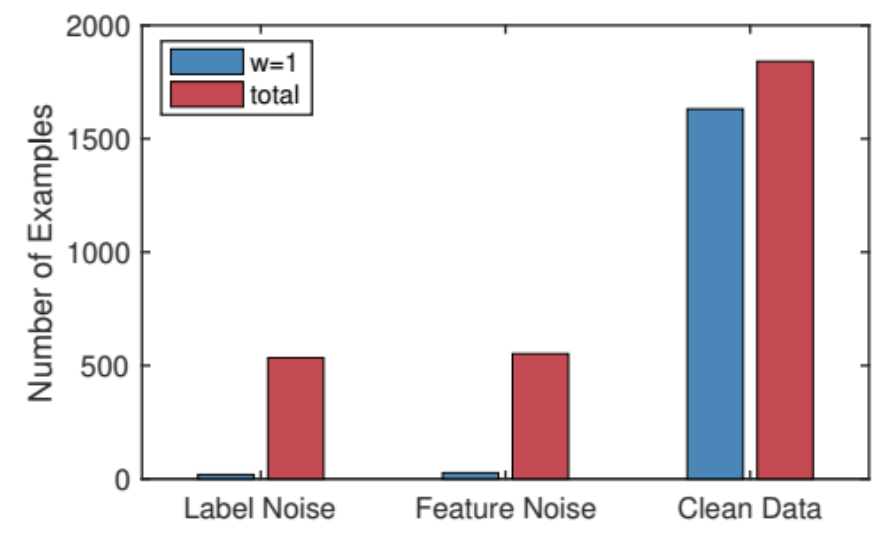
$$w(\mathbf{x}_i^s) = \mathbb{1}\left(\ell_i + \lambda\tau_i \leqslant \gamma\right) \text{ where}$$
$$\ell_i = L_y\left(\mathbf{y}_i^s, G_y\left(G_f\left(\mathbf{x}_i^s\right)\right)\right),$$
$$\tau_i = -\log\left(1 - G_d(G_f(\mathbf{x}_i^s))\right)$$

Table 3: Accuracy on **Office-31** with 40% Mixed Corruption

Method	Office-31 Mixed Corruption						
	A→W	W→A	A→D	D→A	W→D	D→W	Avg
TCL-adversarial_w	85.9	62.9	82.1	**64.9**	97.4	92.5	81.0
TCL-classifier_w	77.3	63.5	80.5	61.2	96.2	91.5	78.4
TCL-easiness	74.0	63.6	77.3	61.9	96.4	90.4	77.3
TCL-transferability	84.7	63.8	83.1	62.6	97.8	92.2	80.7
TCL	**87.4**	**64.6**	**83.1**	62.2	**99.0**	**92.7**	**81.5**

Experiment


0.9
0.8
0.7
0.6
0.5
0.4
Accuracy
0
0.2
0.4
0.6
0.8
Noise level
ResNet
SPL
MentorNet
DAN
RTN
DANN
ADDA
TCL


2000
1500
1000
500
0
Number of Examples
w=1
total
Label Noise
Feature Noise
Clean Data

Summary

+ Applying curriculum learning to domain adaption to against noise.

+ Considering the transferability of examples.

+ Solid experiment

- Do not have an unified objective function