Few-Shot Adversarial Domain Adaptation

Saeid Motiian, Quinn Jones, Seyed Mehdi Iranmanesh, Gianfranco Doretto
Lane Department of Computer Science and Electrical Engineering
West Virginia University
{samotian, qjones1, seiranmanesh, gidoretto}@mix.wvu.edu

NIPS-2017

CONTENTS

1. Introduction

2. Motivation

3. Methods

4. Experiments

Introduction

Source Domain

Train

Model

Adapt

Target Domain

Example

Domain Adaptation

Source Domain $\sim P_S(X, Y)$ $\neq$ Target Domain $\sim P_T(Z, H)$

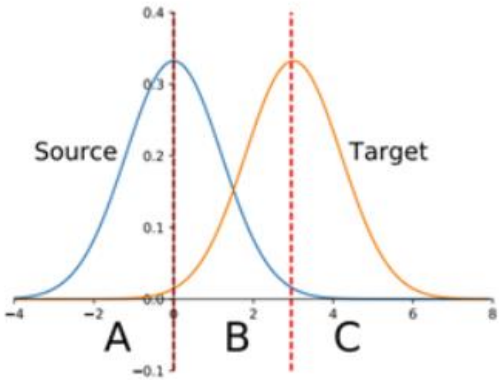
lots of **labeled** data

unlabeled or limited labels

$$D_S = \{(\mathbf{x}_i, y_i), \forall i \in \{1, \ldots, N\}\}$$

$$D_T = \{(\mathbf{z}_j, ?), \forall j \in \{1, \ldots, M\}\}$$

Domain Adaptation



Introduction

- Most domain adaptation approaches try to find a feature space such that the **confusion** between **source** and **target** distributions in that **space** is maximum (domain confusion).

Generative Adversarial Network

Domain Adaptation

Feature Extractor

Discriminator

- By training against the discriminantor, the feature extractor alines the source domain and target domain data distribution in the **feature space**.

Motivation

1. Unsupervised domain adaptation (UDA) algorithms do not need any **target** data labels, but they require **large** amounts of target training samples, which may not always be available.

2. Supervised domain adaptation (SDA) algorithms do require labeled **target** data, and because **labeling** information is available, for the same quantity of **target** data, **SDA** outperforms **UDA**.

3. We aim at handling cases where there is only **one** target labeled sample, and there can even be some classes with **no** target samples at all.

Methods

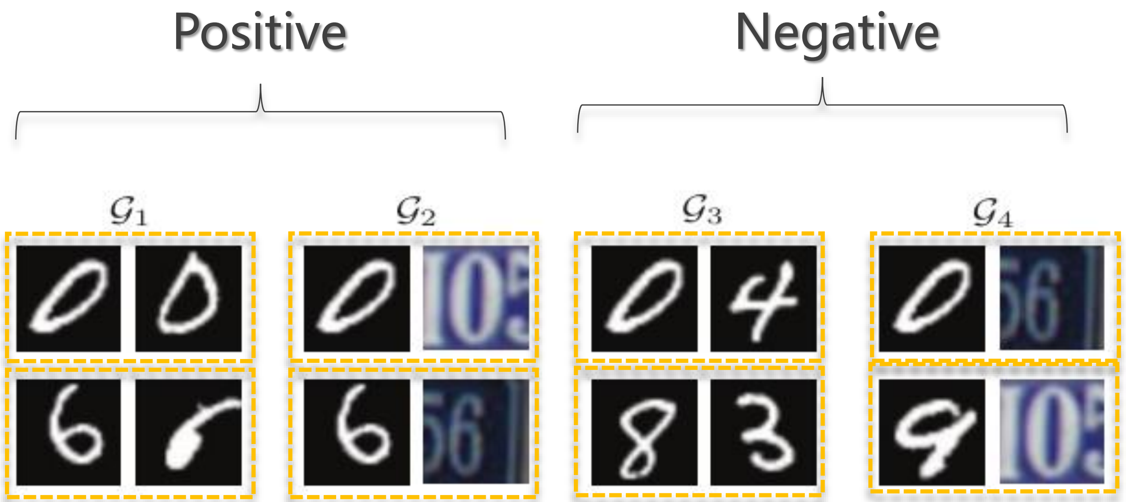


Figure 1: **Examples from MNIST [32] and SVHN [40] of grouped sample pairs.** $\mathcal{G}_1$ is composed of samples of the same class from the source dataset in this case MNIST. $\mathcal{G}_2$ is composed of samples of the same class, but one is from the source dataset and the other is from the target dataset. In $\mathcal{G}_3$ the samples in each pair are from the source dataset but with differing class labels. Finally, pairs in $\mathcal{G}_4$ are composed of samples from the target and source datasets with differing class labels.

$$f = h \circ g$$

$$g : \mathcal{X} \to \mathcal{Z}$$

$$h : \mathcal{Z} \to \mathcal{Y}$$

$$\begin{cases} f_s = h_s \circ g_s \\ f_t = h_t \circ g_t \end{cases}$$

Methods

- If g_s and g_t are able to embed source and target samples, respectively, to a domain invariant space, it is safe to assume from the feature to the label space that $h_t = h_s = h$.

Methods

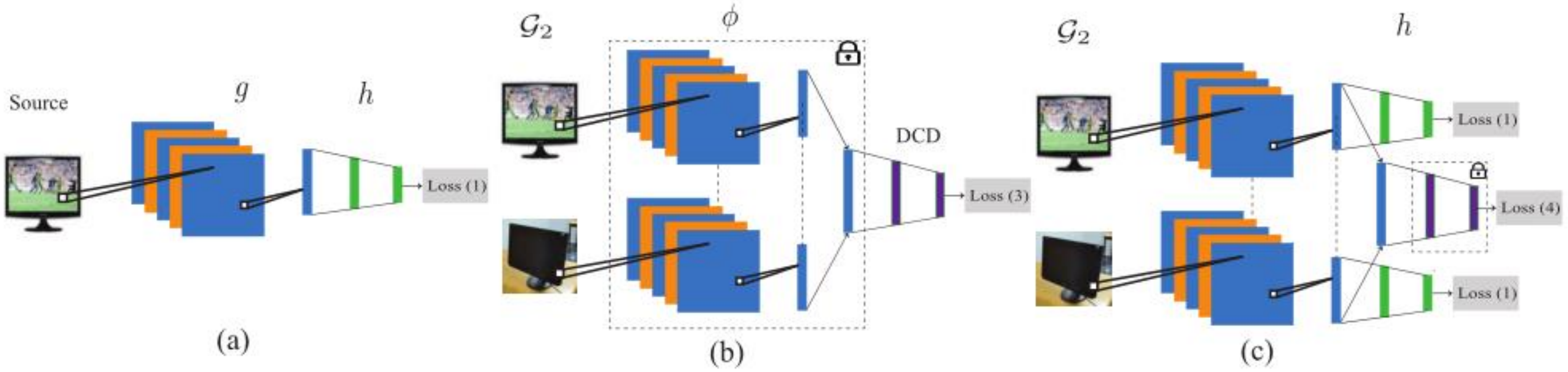


Figure 2: **Few-shot adversarial domain adaptation.** For simplicity we show our networks in the case of weight sharing ($g_s = g_t = g$). **(a)** In the first step, we initialized g and h using the source samples $\mathcal{D}_s$. **(b)** We freeze g and train a DCD. The picture shows a pair from the second group $\mathcal{G}_2$ when the samples come from two different distributions but the same class label. **(c)** We freeze the DCD and update g and h.

$$\mathcal{L}_C(f) = E[\ell(f(X^s), Y)] \ , \tag{1}$$

$$\mathcal{L}_{FADA-D} = -E[\sum_{i=1}^{4} y_{\mathcal{G}_i} \log(D(\phi(\mathcal{G}_i)))] \ , \tag{3}$$

$$\mathcal{L}_{FADA-g} = -\gamma E[y_{\mathcal{G}_1} \log(D(g(\mathcal{G}_2))) - y_{\mathcal{G}_3} \log(D(g(\mathcal{G}_4)))] + E[\ell(f(X^s), Y)] + E[\ell(f(X^t), Y)] \ , \tag{5}$$

Methods

Algorithm 1 FADA algorithm

Train g and h on $\mathcal{D}_s$ using (1).
Uniformly sample $\mathcal{G}_1$,$\mathcal{G}_3$ from $\mathcal{D}_s$x$\mathcal{D}_s$.
Uniformly sample $\mathcal{G}_2$,$\mathcal{G}_4$ from $\mathcal{D}_s$x$\mathcal{D}_t$.
Train DCD w.r.t. $g_t = g_s = g$ using (3).
while not convergent **do**
    Update g and h by minimizing (5).
    Update DCD by minimizing (3).
end while

Experiments

Table 1: **MNIST-USPS-SVHN datasets.** Classification accuracy for domain adaptation over the MNIST, USPS, and SVHN datasets. $\mathcal{M}$, $\mathcal{U}$, and $\mathcal{S}$ stand for MNIST, USPS, and SVHN domain. `LB` is our base model without adaptation. `FT` and `FADA` stand for fine-tuning and our method, respectively.

		Traditional UDA			Adversarial UDA										
	LB	[60]	[45]	[15]	[33]	[59]	[49]	SDA	1	2	3	4	5	6	7
$\mathcal{M} \to \mathcal{U}$	65.4	47.8	60.7	91.8	91.2	89.4	92.5	FT	82.3	84.9	85.7	86.5	87.2	88.4	88.6
								[38]	85.0	89.0	90.1	91.4	92.4	93.0	92.9
								FADA	89.1	91.3	91.9	93.3	93.4	94.0	**94.4**
$\mathcal{U} \to \mathcal{M}$	58.6	63.1	67.3	73.7	89.1	90.1	90.8	FT	72.6	78.2	81.9	83.1	83.4	83.6	84.0
								[38]	78.4	82.2	85.8	86.1	88.8	89.6	89.4
								FADA	81.1	84.2	87.5	89.9	91.1	91.2	**91.5**
$\mathcal{S} \to \mathcal{M}$	60.1	-	-	82.0	76.0	-	84.7	FT	65.5	68.6	70.7	73.3	74.5	74.6	75.4
								FADA	72.8	81.8	82.6	85.1	86.1	86.8	**87.2**
$\mathcal{M} \to \mathcal{S}$	20.3	-	-	40.1	-	-	36.4	FT	29.7	31.2	36.1	36.7	38.1	38.3	39.1
								FADA	37.7	40.5	42.9	46.3	46.1	46.8	**47.0**
$\mathcal{S} \to \mathcal{U}$	66.0	-	-	-	-	-	-	FT	69.4	71.8	74.3	76.2	78.1	77.9	78.9
								FADA	78.3	83.2	85.2	85.7	86.2	87.1	**87.5**
$\mathcal{U} \to \mathcal{S}$	15.3	-	-	-	-	-	-	FT	19.9	22.2	22.8	24.6	25.4	25.4	25.6
								FADA	27.5	29.8	34.5	36.0	37.9	41.3	**42.9**

Experiments

Table 2: **Office dataset.** Classification accuracy for domain adaptation over the 31 categories of the Office dataset. $\mathcal{A}$, $\mathcal{W}$, and $\mathcal{D}$ stand for Amazon, Webcam, and DSLR domain. LB is our base model without adaptation. Since we do not train any convolutional layers and only use pre-computed DeCaF-fc7 features as input, we expect a more challenging task compared to [58, 27].

	Unsupervised Methods				Supervised Methods			
	LB	[60]	[34]	[15]	[58]	[27]	[38]	FADA
$\mathcal{A} \to \mathcal{W}$	61.2 ± 0.9	61.8 ± 0.4	68.5 ± 0.4	68.7 ± 0.3	82.7 ± 0.8	84.5 ± 1.7	$\mathbf{88.2 \pm 1.0}$	88.1 ± 1.2
$\mathcal{A} \to \mathcal{D}$	62.3 ± 0.8	64.4 ± 0.3	67.0 ± 0.4	67.1 ± 0.3	86.1 ± 1.2	86.3 ± 0.8	$\mathbf{89.0 \pm 1.2}$	88.2 ± 1.0
$\mathcal{W} \to \mathcal{A}$	51.6 ± 0.9	52.2 ± 0.4	53.1 ± 0.3	54.09 ± 0.5	65.0 ± 0.5	65.7 ± 1.7	$\mathbf{72.1 \pm 1.0}$	71.1 ± 0.9
$\mathcal{W} \to \mathcal{D}$	95.6 ± 0.7	98.5 ± 0.4	$\mathbf{99.0 \pm 0.2}$	$\mathbf{99.0 \pm 0.2}$	97.6 ± 0.2	97.5 ± 0.7	97.6 ± 0.4	97.5 ± 0.6
$\mathcal{D} \to \mathcal{A}$	58.5 ± 0.8	52.1 ± 0.8	54.0 ± 0.4	56.0 ± 0.5	66.2 ± 0.3	66.5 ± 1.0	$\mathbf{71.8 \pm 0.5}$	68.1 ± 06
$\mathcal{D} \to \mathcal{W}$	80.1 ± 0.6	95.0 ± 0.5	96.0 ± 0.3	$\mathbf{96.4 \pm 0.3}$	95.7 ± 0.5	95.5 ± 0.6	$\mathbf{96.4 \pm 0.8}$	$\mathbf{96.4 \pm 0.8}$
Average	68.2	70.6	72.9	73.6	82.2	82.6	**85.8**	84.9

Experiments

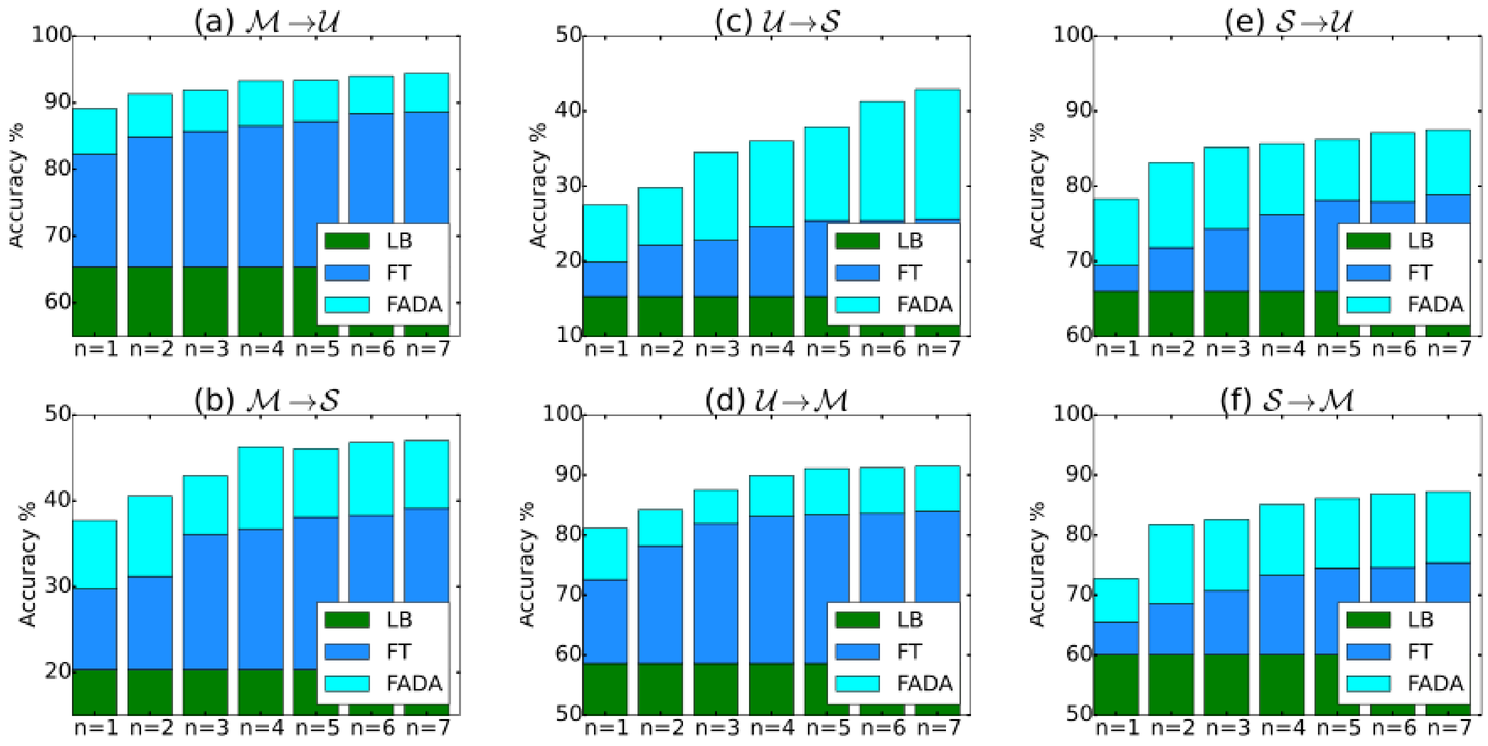


Figure 3: **MNIST-USPS-SVHN summary.** The lower bar of each column represents the `LB` as reported in Table 1 for the corresponding domain pair. The middle bar is the improvement of fine-tuning `FT` the base model using the available target data reported in Table 1. The top bar is the improvement of `FADA` over `FT`, also reported in Table 1.

THANKS FOR WATCHING