Universal Domain Adaptation

(2019)

Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan

Summary

Contributions

In domain adaptation, we almost always suppose that the target and source have a shared label set where the number of classes is the same across dataset, but the reality is a lot more messy that this, since we can have different classes between training and when operating the model, such as using the model on rare species of animals but we only trained on animals that are common.

The authors propose a universal domain adaptation independent on the label set of each domain as illustrated above.

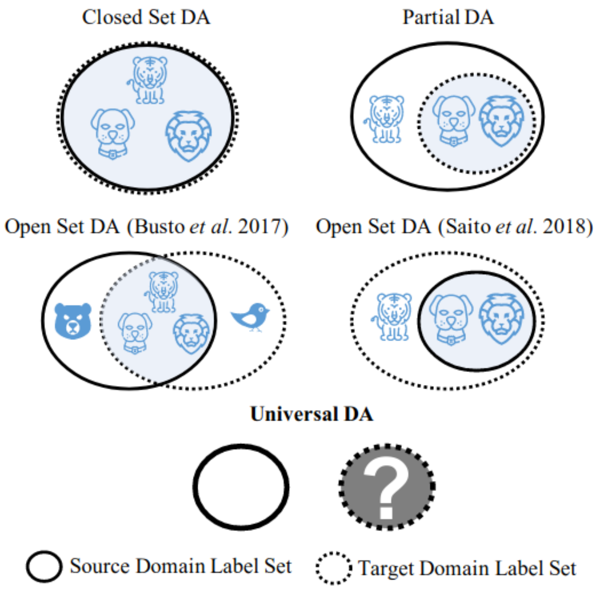


Figure 1. Universal Domain Adaptation (UDA) and existing domain adaptation settings with respect to label sets of source and target domains (blue shades indicate shared labels). Only UDA is able to deal with the setting that the label set of target domain is unknown.

The proposed methods solves two problems, the domain gap, the problem we always try to tackle in unsupervised domain adaptation and category cap, which is not taken into consideration in the majority of cases. The authors do not impose an prior knowledge on the label set of target and source, which is important and intuitive since we do not know the classes of the target domain before hand, and propose a Universal Adaptation Network (UAN) which exploits both the domain similarity and the prediction uncertainty of each sample to develops weighting mechanism for discovering label sets shared by both domains and promote common-class adaptation.

Method

In Universal Domain Adaptation (UDA), a source domain $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}$ consisting of n_s labeled samples and a target domain $\mathcal{D}_t = \{(\mathbf{x}_i^t)\}$ of n_t unlabeled samples are provided at training. Note that the source data are sampled from distribution p while the target data from distribution q. We use C_s to denote the label set of source domain and C_t the label set of target domain. $\mathcal{C} = \mathcal{C}_s \cap \mathcal{C}_t$ is the common label set shared by both domains $\overline{\mathcal{C}}_s = \mathcal{C}_s \backslash \mathcal{C}$ and $\overline{\mathcal{C}}_t = \mathcal{C}_t \backslash \mathcal{C}$ represent the label sets private to the source domain and the target domain respectively. The objective is to design a model capable of producing good results across a wide range of $\xi = \frac{|\mathcal{C}_s \cap \mathcal{C}_t|}{|\mathcal{C}_s \cup \mathcal{C}_t|}$, which measure the commonness of target and source, even without knowing ξ beforehand.

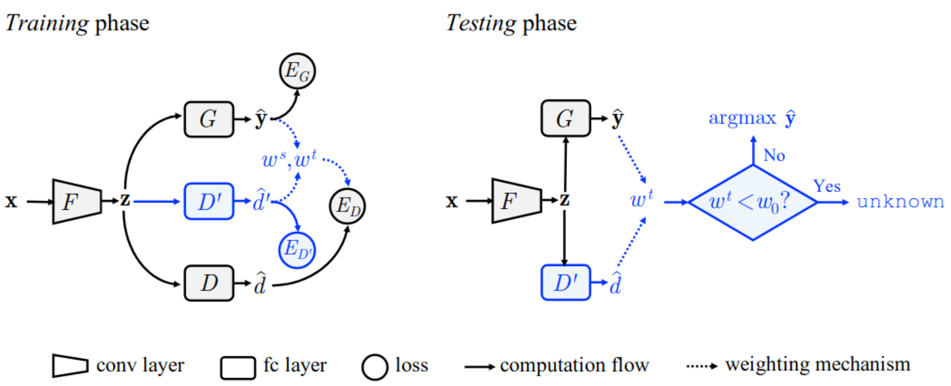


Figure 2. The training and testing phases of the Universal Adaptation Network (UAN) designed for Universal Domain Adaptation (UDA).

The authors propose Universal Adaptation Network (illustrated in the figure above), similar to previous works, the model contain a feature extractor F, a classifier G and a discriminator D, the classifier in trained on top of the features using the labeled source dataset; and the discriminator pushes the features extractor to produce invariance representations across domains. A new component is introduced, a domain similarity classifier D', which tells us how much an examples x is similar to source examples, the loss in this case is as follows:

$$E_G = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p} L(\mathbf{y}, G(F(\mathbf{x})))$$

$$E_{D'} = -\mathbb{E}_{\mathbf{x}\sim p} \log D'(F(\mathbf{x})) - \mathbb{E}_{\mathbf{x}\sim q} \log (1 - D'(F(\mathbf{x})))$$

$$E_D = -\mathbb{E}_{\mathbf{x}\sim p} w^s(\mathbf{x}) \log D(F(\mathbf{x})) - \mathbb{E}_{\mathbf{x}\sim q} w^t(\mathbf{x}) \log(1 - D(F(\mathbf{x})))$$

We have two weights $w^s(\mathbf{x})$ and $w^t(\mathbf{x})$, indicating the probability of example x belonging to the common set $\mathcal{C}$, and the model is trained to minimize the loss of F, G and D' and maximize that of D. The two weights are computed as follows:

$$w^s(\mathbf{x}) = \frac{H(\hat{\mathbf{y}})}{\log |\mathcal{C}_s|} - \hat{d}'(\mathbf{x})$$

$$w^t(\mathbf{x}) = \hat{d}'(\mathbf{x}) - \frac{H(\hat{\mathbf{y}})}{\log |\mathcal{C}_s|}$$

To get the weights, we use both the output d' of the domain similarity classifier D', and the uncertainty of the classifier $H(\hat{\mathbf{y}})$, now for examples that are from source, d' will be high and $H(\hat{\mathbf{y}})$ will be low, especially is we have an example of label not in the target label set, and vice-versa. So in x is from source, $w^s(\mathbf{x})$ is negative (close to -1) and $w^t(\mathbf{x})$ if positive and close to 1, and in the opposite case $w^t(\mathbf{x})$ is close to -1, so we can use the weights during increase the loss of the discriminator, but also during testing, since is $w^t(\mathbf{x})$ is quite low, the example is likely to not be very similar to any examples in the source dataset, and we need to outputs and unknown class:

$$y(\mathbf{x}) = \begin{cases} \text{unknown} & w^t < w_0 \\ \text{argmax}(\hat{\mathbf{y}}) & w^t \geq w_0 \end{cases}$$

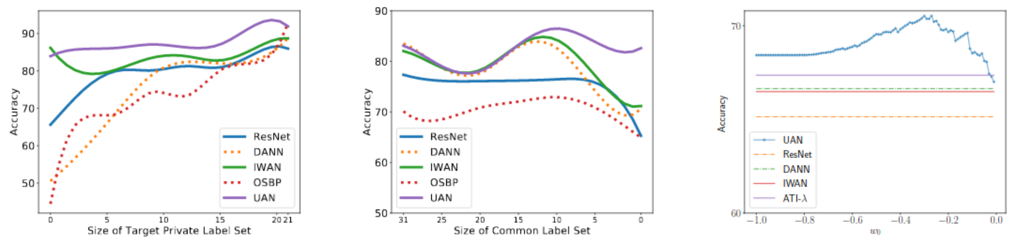
Results

Table 1. Average class accuracy (%) of universal domain adaptation tasks on **Office-Home** ($\xi = 0.15$) dataset (ResNet)

Method	Office-Home												
	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
ResNet [13]	59.37	76.58	87.48	69.86	71.11	81.66	73.72	56.30	86.07	78.68	59.22	78.59	73.22
DANN [6]	56.17	81.72	86.87	68.67	73.38	83.76	69.92	56.84	85.80	79.41	57.26	78.26	73.17
RTN [23]	50.46	77.80	86.90	65.12	73.40	85.07	67.86	45.23	85.50	79.20	55.55	78.79	70.91
IWAN [45]	52.55	81.40	86.51	70.58	70.99	85.29	74.88	57.33	85.07	77.48	59.65	78.91	73.39
PADA [45]	39.58	69.37	76.26	62.57	67.39	77.47	48.39	35.79	79.60	75.94	44.50	78.10	62.91
ATI [28]	52.90	80.37	85.91	71.08	72.41	84.39	74.28	57.84	85.61	76.06	60.17	78.42	73.29
OSBP [35]	47.75	60.90	76.78	59.23	61.58	74.33	61.67	44.50	79.31	70.59	54.95	75.18	63.90
UAN w/o d	61.60	81.86	87.67	74.52	73.59	84.88	73.65	57.37	86.61	81.58	62.15	79.14	75.39
UAN w/o y	56.63	77.51	87.61	71.96	69.08	83.18	71.40	56.10	84.24	79.27	60.59	78.35	72.91
UAN	**63.00**	**82.83**	**87.85**	**76.88**	**78.70**	**85.36**	**78.22**	**58.59**	**86.80**	**83.37**	**63.17**	**79.43**	**77.02**

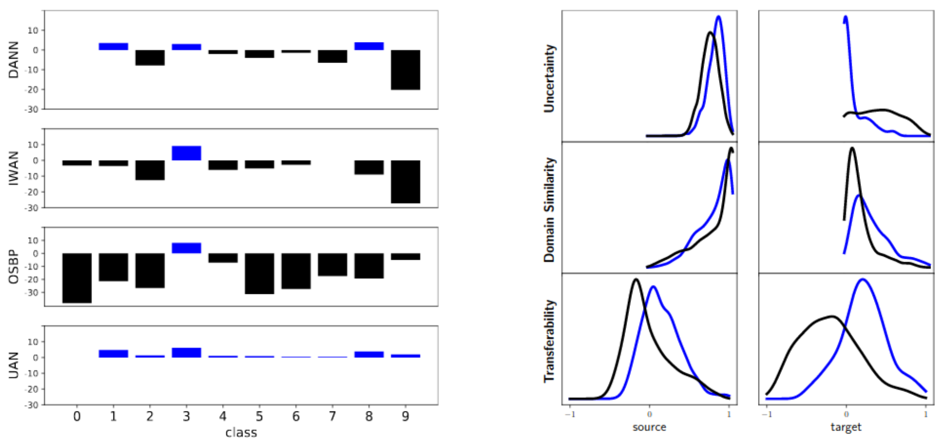
Table 2. Average class accuracy (%) on **Office-31** ($\xi = 0.32$), **ImageNet-Caltech** ($\xi = 0.07$) and **VisDA2017** ($\xi = 0.50$) (ResNet)

Method	Office-31							ImageNet-Caltech		VisDA
	A → W	D → W	W → D	A → D	D → A	W → A	Avg	I → C	C → I	
ResNet [13]	75.94	89.60	90.91	80.45	78.83	81.42	82.86	70.28	65.14	52.80
DANN [6]	80.65	80.94	88.07	82.67	74.82	83.54	81.78	71.37	66.54	52.94
RTN [23]	85.70	87.80	88.91	82.69	74.64	83.26	84.18	71.94	66.15	53.92
IWAN [45]	85.25	90.09	90.00	84.27	84.22	86.25	86.68	72.19	66.48	58.72
PADA [45]	85.37	79.26	90.91	81.68	55.32	82.61	79.19	65.47	58.73	44.98
ATI [28]	79.38	92.60	90.08	84.40	78.85	81.57	84.48	71.59	67.36	54.81
OSBP [35]	66.13	73.57	85.62	72.92	47.35	60.48	67.68	62.08	55.48	30.26
UAN	**85.62**	**94.77**	**97.99**	**86.50**	**85.45**	**85.12**	**89.24**	**75.28**	**70.17**	**60.83**



(a) Accuracy w.r.t. $|\overline{\mathcal{C}}_t|$ (b) Accuracy w.r.t. $|\mathcal{C}|$ (c) Sensitivity to w_0

Figure 3. (a) Accuracy w.r.t. $|\overline{\mathcal{C}}_t|$ in task **A** → **D**, $\xi = 0.32$. (b) Accuracy w.r.t. $|\mathcal{C}|$ in task **A** → **D**. (c) Performance w.r.t. threshold w_0.



(a) Negative Transfer in UDA (b) Hypotheses Quality (blue for *common* and black for *private*)

Figure 4. (a) The negative transfer influence in UDA (task **Ar** → **Cl**). (b) Justification of validity of hypotheses in Section 3.4.