ICPR2020

Supervised Domain Adaptation using Graph Embedding

Lukas Hedegaard Morsing*, Omar Ali Sheikh-Omar*, and Alexandros Iosifidis
Department of Electrical and Computer Engineering, Aarhus University, Denmark
lh@eng.au.dk, sheikhomar@mailbox.org, ai@eng.au.dk

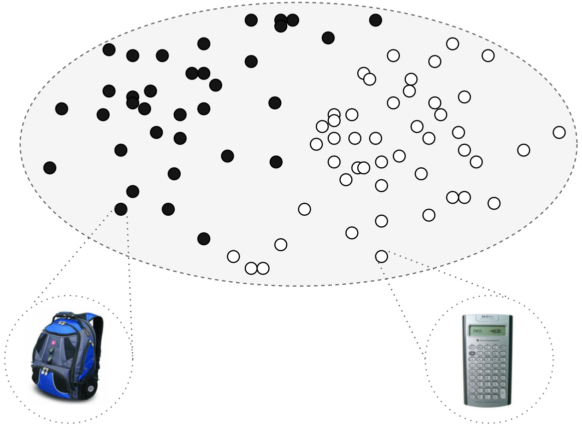
AARHUS UNIVERSITY

open Deep Robotics

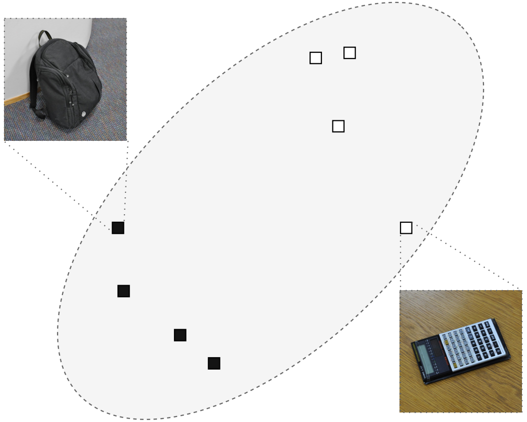
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871449

Transfer Learning

- Leveraging data from a *Source Domain* to improve performance on a *Target Domain*
- Domain $\mathcal{D}$: Feature-space $\mathcal{X}$ Marginal probability distribution $p(\mathbf{X})$
- Task $\mathcal{T}$: Label-space $\mathcal{Y}$ Predictive function $f(\cdot)$



Source S



Target T

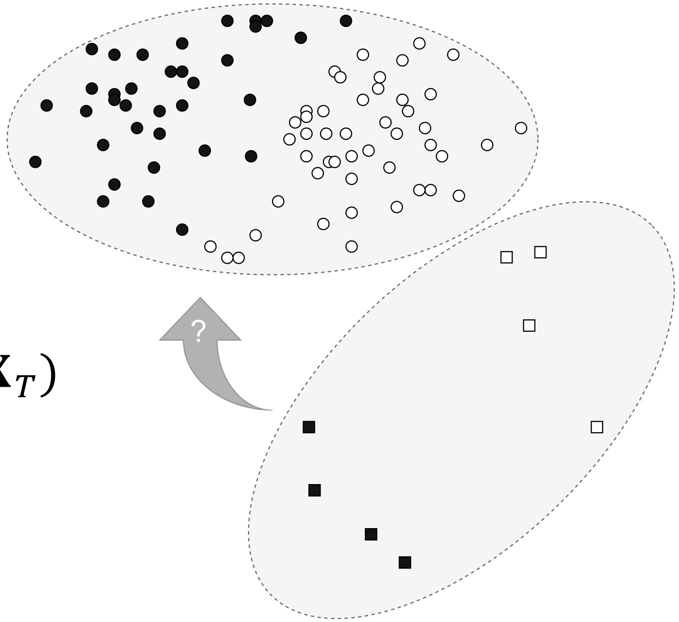
Domain Adaptation

Assumptions:

- Same feature space and label-space, i.e.

  $\mathcal{X}_S = \mathcal{X}_T \qquad \mathcal{Y}_S = \mathcal{Y}_T$

- Misalignment of source and target datasets
- Domain shift is a *covariate shift*: $p(\mathbf{X}_S) \neq p(\mathbf{X}_T)$



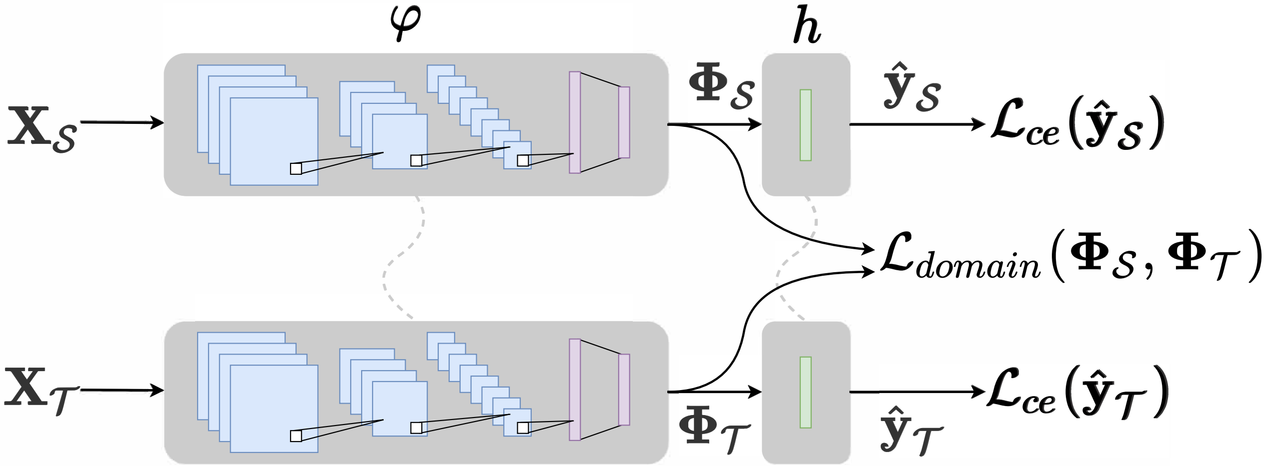
Deep two-stream network for DA

- Deep Neural Network: $f(\mathbf{x}) = h(\varphi(\mathbf{x}))$

  Feature-extractor $\varphi: \mathcal{X} \rightarrow \mathcal{Z}$ (latent space) *Classifier* $h: \mathcal{Z} \rightarrow \mathcal{Y}$ (label space)

- Add penalty on latent space representation difference: $\mathcal{L}_{domain}(\cdot)$

  $$\arg\min_{\theta_\varphi, \theta_h} \mathcal{L}_{domain} + \beta \mathcal{L}_{ce}^{S} + \gamma \mathcal{L}_{ce}^{T}$$



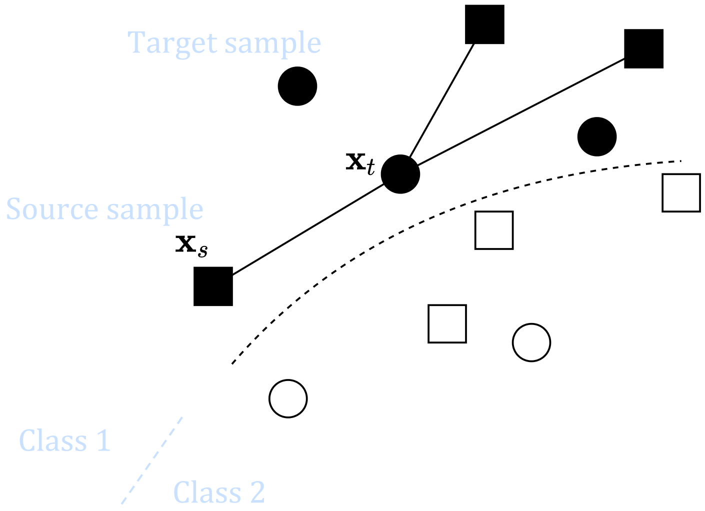
Related Works - CCSA

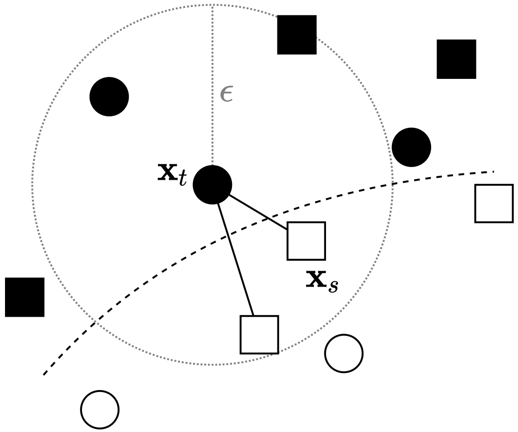
“Classification and Contrastive Semantic Alignment Loss” [1]

$$\mathcal{L}_{CSA} = \sum_{s \in \mathcal{D}_S} \sum_{t \in \mathcal{D}_T} \alpha \frac{1}{2} d(\mathbf{x}_s, \mathbf{x}_t)^2 - (1-\alpha) \frac{1}{2} \max\{0, 1 - d(\mathbf{x}_s, \mathbf{x}_t)\}^2$$

$$\alpha = \begin{cases} 1, & \ell_S = \ell_T \\ 0, & \ell_S \neq \ell_T \end{cases}$$

$$d(\mathbf{x}_s, \mathbf{x}_t) = \|\mathbf{x}_s - \mathbf{x}_t\|_2$$



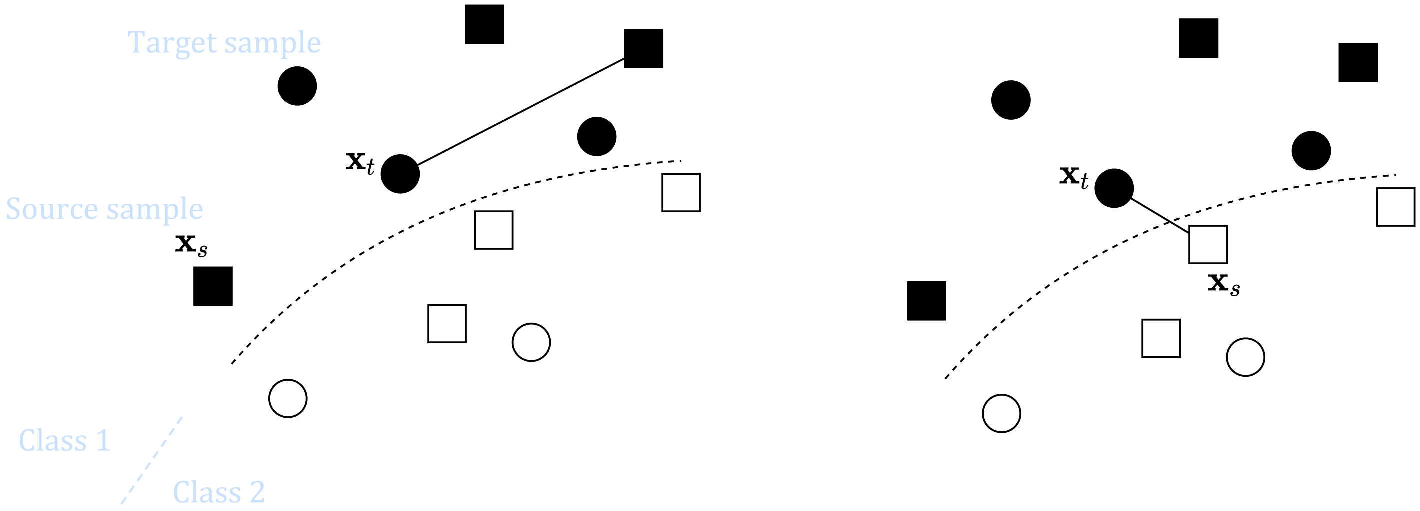


[1] S. Motiian, M. Piccirilli, D. A. Adjeroh, and G. Doretto, “Unified deep supervised domain adaptation and generalization,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5715–5725

Related Works – *d*-SNE

"Domain Adaptation using Stochastic Neighbourhood Embedding" [2]

$$\mathcal{L}_{d-\mathrm{SNE}} = \sum_{t \in \mathcal{D}_T} \sup_{s \in \mathcal{D}_S^{(\ell_s = \ell_t)}} \{a \mid a \in d(\mathbf{x}_s, \mathbf{x}_t)\} - \inf_{s \in \mathcal{D}_S^{(\ell_s \neq \ell_t)}} \{b \mid b \in d(\mathbf{x}_s, \mathbf{x}_t)\} \qquad d(\mathbf{x}_s, \mathbf{x}_t) = \|\mathbf{x}_s - \mathbf{x}_t\|_2$$



[2] X. Xu, X. Zhou, R. Venkatesan, G. Swaminathan, and O. Majumder, "d-sne: Domain adaptation using stochastic neighborhood embedding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2497–2506.

AARHUS UNIVERSITY

Domain Adaptation via Graph Embedding

Graph preserving criterion $$\sum_{i=1}^{N}\sum_{j=1}^{N}\left\|\mathbf{\Phi}^{(i)}-\mathbf{\Phi}^{(j)}\right\|_2^2\,\mathbf{W}^{(i,j)}=\mathrm{Tr}(\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}^{\top})$$

Features	$\mathbf{\Phi}=[\varphi(\mathbf{X}_S)\quad \varphi(\mathbf{X}_T)]$
Weight matrix	$\mathbf{W}$
Degree matrix	$\mathbf{D}=\sum_j \mathbf{W}^{(i,j)}$
Laplacian matrix	$\mathbf{L}=\mathbf{D}-\mathbf{W}$
Penalty —||—	$\mathbf{B}=\mathbf{D}_p-\mathbf{W}_p$

Within-class spread

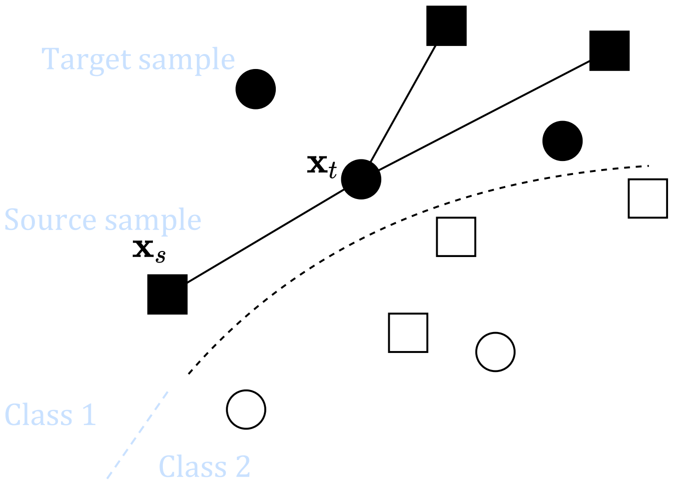
$$\mathcal{L}_{\mathrm{DAGE}}=\frac{\mathrm{Tr}(\mathbf{\Phi}\mathbf{L}\mathbf{\Phi}^{\top})}{\mathrm{Tr}(\mathbf{\Phi}\mathbf{B}\mathbf{\Phi}^{\top})}$$

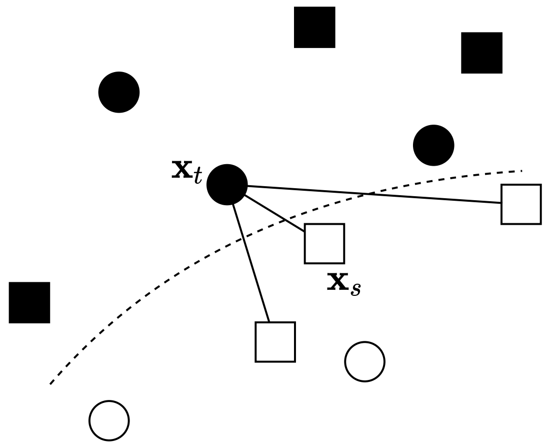
Between-class spread

AARHUS UNIVERSITY

DAGE-LDA

Inspired by Linear Discriminant Analysis





$$\mathbf{W}^{(i,j)} = \begin{cases} 1, & \ell_i = \ell_j \wedge \mathcal{D}_i \neq \mathcal{D}_j \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbf{W}_p^{(i,j)} = \begin{cases} 1, & \ell_i \neq \ell_j \wedge \mathcal{D}_i \neq \mathcal{D}_j \\ 0, & \text{otherwise} \end{cases}$$

Experiment setup

Comparison between published methods is not trivial

- Difference in network implementation → Re-implementation using publicly available code
- Different hyperparameter search methods
- Different computational budget → Bayesian Optimisation with same computational budget
- Differences in data sampling procedure

→ Exact same training pipeline

Results – Office31



Experimental setup

- Pairwise transfer among three domains
- A subset of data is sampled for each run
- Five runs are performed

Domain	**Amazon** ($\mathcal{A}$)	**DSLR** ($\mathcal{D}$)	**Webcam** ($\mathcal{W}$)
Source samples/class	20	8	8
Target samples/class	3	3	3
Classes	31	31	31

TABLE I: Office31 dataset statistics for experiment

	$\mathcal{A} \rightarrow \mathcal{D}$	$\mathcal{A} \rightarrow \mathcal{W}$	$\mathcal{D} \rightarrow \mathcal{A}$	$\mathcal{D} \rightarrow \mathcal{W}$	$\mathcal{W} \rightarrow \mathcal{A}$	$\mathcal{W} \rightarrow \mathcal{D}$	Avg.
FT-Source	66.6 ± 3.0	59.8 ± 2.1	42.8 ± 5.2	92.3 ± 2.8	44.0 ± 0.7	98.5 ± 1.2	67.4
FT-Target	71.4 ± 2.0	74.0 ± 4.9	56.2 ± 3.6	95.9 ± 1.2	50.2 ± 2.6	99.1 ± 0.8	74.5
CCSA	84.8 ± 2.1	87.5 ± 1.5	$\mathbf{66.5 \pm 1.9}$	97.2 ± 0.7	64.0 ± 1.6	98.6 ± 0.4	83.1
d-SNE	$\mathbf{86.5 \pm 2.5}$	$\mathbf{88.7 \pm 1.9}$	65.9 ± 1.1	97.6 ± 0.7	63.9 ± 1.2	99.0 ± 0.5	**83.6**
DAGE-LDA	85.9 ± 2.8	87.8 ± 2.3	66.2 ± 1.4	$\mathbf{97.9 \pm 0.6}$	$\mathbf{64.2 \pm 1.2}$	$\mathbf{99.5 \pm 0.5}$	**83.6**

TABLE II: Macro average classification accuracy (%) for Office-31 using a VGG16 network pretrained on ImageNet. The reported results are the mean and standard deviation across five runs.

Results – MNIST → USPS

Experimental setup

- Source data: MNIST with 2000 samples / class
- Target data: USPS with varying samples / class
- A subset of data is sampled for each run
- Ten runs are performed

Samples/class	1	3	5	7
CCSA	$\mathbf{75.6 \pm 2.1}$	$\mathbf{85.0 \pm 1.4}$	87.8 ± 0.7	89.1 ± 0.7
d-SNE	69.0 ± 1.7	80.4 ± 1.7	86.1 ± 0.9	87.7 ± 0.9
DAGE-LDA	67.0 ± 1.9	82.7 ± 1.7	$\mathbf{89.0 \pm 0.8}$	$\mathbf{90.7 \pm 0.5}$

TABLE III: Classification accuracy (%) for MNIST → USPS with a varying number of target samples per class.

In conclusion

- Re-evaluation of prior state-of-the methods in a fair comparison
- Novel use of Graph Embedding: trace-ratio objective as loss in deep NN
- A simple LDA-inspired Domain Adaptation loss, DAGE-LDA

  matches or beats the overall accuracy or prior state-of-the-art methods

Thank You for Your attention