Lifelong Domain Adaptation via Consolidated Internal Distribution

Mohammad Rostami

USC Information Sciences Institute

Motivation

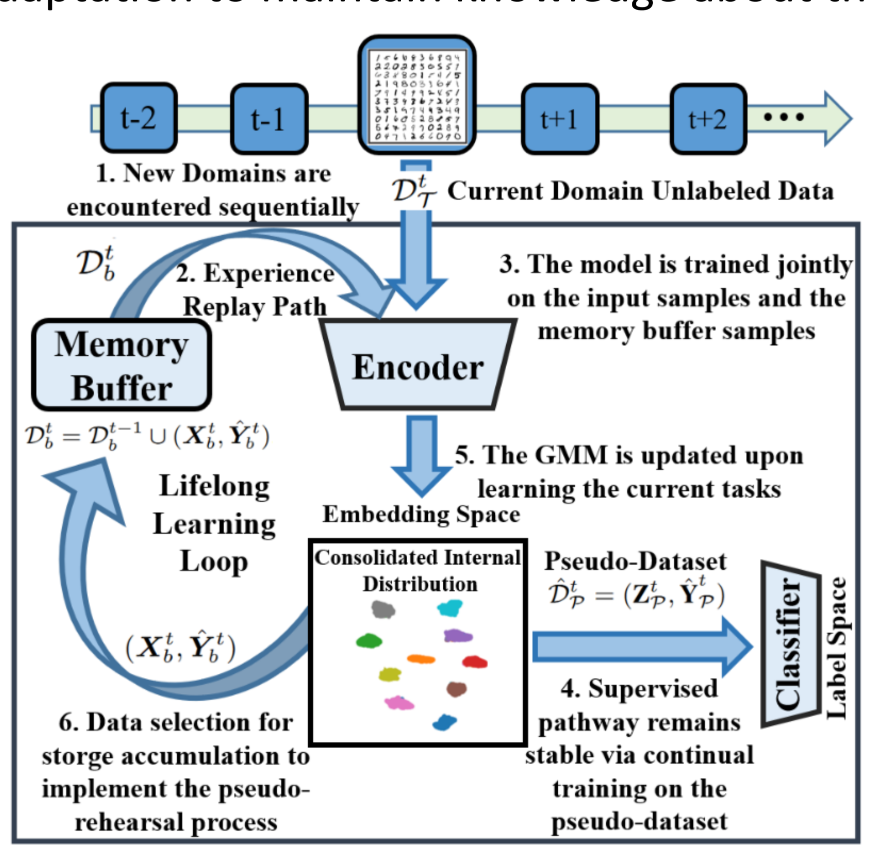
- Distributional domain shift is a dynamic and continual process in practice which leads to model performance degradation during execution time after an initial training phase on a source domain
- Existing unsupervised domain adaptation (UDA) algorithms mostly address domain shift for a single target domain using a source domain
- Existing methods mostly require joint access to the source domain annotated data to and the target domain unannotated data
- Catastrophic forgetting is a challenge for neural networks, when updating a neural network leads to forgetting what it has learned before

Objectives

- We model the UDA problem in a continual learning setting, where the model is continually updated during model execution.
- We enable updating a trained model for a source domain to tackle domain shift continually without full access to the prior domain data
- We consolidate the internally learned distribution by the model to enable knowledge transfer across the domain
- We benefit from experience replay to tackle catastrophic forgetting

Proposed Framework

- We model is trained on an initial source domain with annotated data
- The internally learned distribution for classification of input data, modeled by network responses in a hidden layer, is estimated as a Gaussian distribution and used to store a subset of samples in a buffer
- During model execution, it is updated when it encounters target domain with unannotated data while consolidating the input distribution
- Samples that are stored in the memory buffer are replayed back during model adaptation to maintain knowledge about the past domains



Lifelong Domain Adaptation Agent

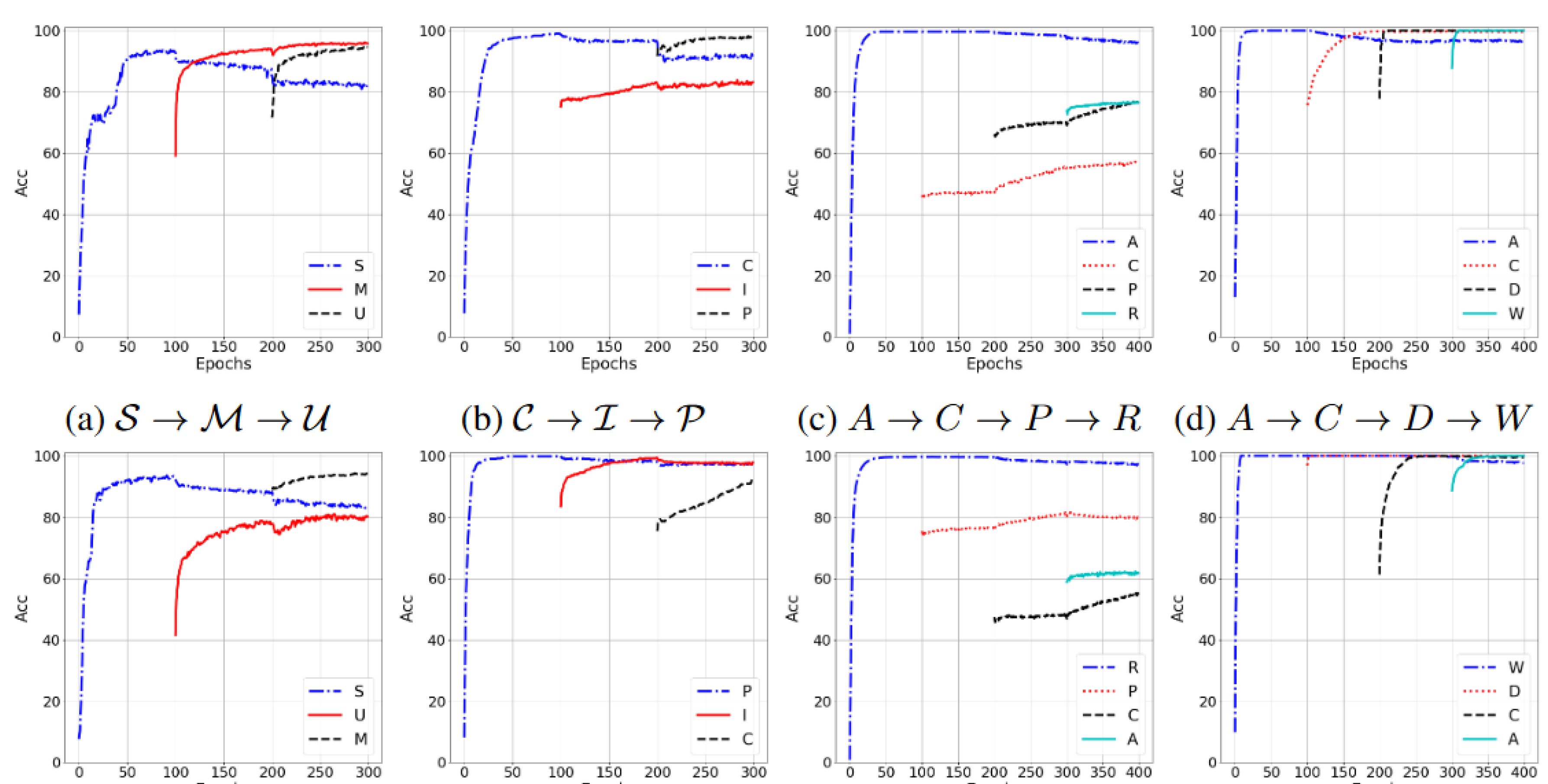
Internal Distribution for Lifelong Domain Adaptation

- We learn the internally learned distribution as a multimodal Gaussian distribution in an embedding modeled by the output of an encoder
- We consolidate the internal distribution
- We use the internal distribution to select samples that are stored in the memory buffer
- We update the model by solving the following opt

$$\min_{v,w} \sum_{i=1}^{N} \mathcal{L}(h_w(z_i^p), \hat{y}_i^{p,t}) + \sum_{i=1}^{N_b} \mathcal{L}(h_w(\phi_v(x_i^b)), \hat{y}_i^b)$$
$$+ \lambda D(\phi_v(p_t(\boldsymbol{X}_t)), \hat{p}_J^t(\boldsymbol{Z}_{\mathcal{P}}^t)) + \lambda D(\phi_v(p_t(\boldsymbol{X}_b^t)), \hat{p}_J^t(\boldsymbol{Z}_{\mathcal{P}}^t))$$

Results

- We use the standard UDA benchmark datasets: digit recognition, ImageCLEF, Office-Home, and Office-Caltech
- Experiments are performed in sequential UDA setting
- Comparison is performed existing UDA methods in binary UDA setting which can be used as an upperbound the effectiveness of our method
- Classification accuracy on the target domain is used for comparison and learning curves are generated to study LL
- A pretrained backbone encoder is used for each dataset according to the precedent in the literature for fair comparison



(a) $\mathcal{S} \to \mathcal{M} \to \mathcal{U}$ (b) $\mathcal{C} \to \mathcal{I} \to \mathcal{P}$ (c) $A \to C \to P \to R$ (d) $A \to C \to D \to W$

(e) $\mathcal{S} \to \mathcal{U} \to \mathcal{M}$ (f) $\mathcal{P} \to \mathcal{I} \to \mathcal{C}$ (g) $R \to P \to C \to A$ (h) $W \to D \to C \to A$

Learning curves for sequential UDA tasks on (a,e) digit, (b,f) ImageClef, (c,g) Office-Home, and (d,h) Office-Caltech datasets. (Best viewed in color).

Method	$\mathcal{M} \to \mathcal{U}$	$\mathcal{U} \to \mathcal{M}$	$\mathcal{S} \to \mathcal{M}$	Method	$\mathcal{M} \to \mathcal{U}$	$\mathcal{U} \to \mathcal{M}$	$\mathcal{S} \to \mathcal{M}$
GtA [10]	92.8 ± 0.9	90.8 ± 1.3	92.4 ± 0.9	CDAN [13]	93.9	96.9	88.5
CoGAN [46]	91.2 ± 0.8	89.1 ± 0.8	-	SHOT [47]	89.6±5.0	96.8±0.4	91.9±0.4
ADDA [3]	89.4 ± 0.2	90.1 ± 0.8	76.0 ± 1.8	CyCADA [48]	95.6 ± 0.2	96.5 ± 0.1	90.4 ± 0.4
RevGrad [15]	77.1 ± 1.8	73.0 ± 2.0	73.9	JDDA [44]	-	97.0 ±0.2	93.1 ±0.2
DRCN [21]	91.8 ± 0.1	73.7 ± 0.4	82.0 ± 0.2	OPDA [27]	70.0	60.2	-
ETD [49]	96.4± 0.3	96.3± 0.1	**97.9**± 0.4	MML [50]	77.9	60.5	62.9
Source Only	90.1±2.6	80.2±5.7	67.3±2.6	LDAuCID	**96.8** ± 0.2	**98.4** ± 0.1	91.4 ± 2.2

Table 1: Classification accuracy for UDA tasks between MNIST, USPS, and SVHN datasets.

Method	$\mathcal{I} \to \mathcal{P}$	$\mathcal{P} \to \mathcal{I}$	$\mathcal{I} \to \mathcal{C}$	$\mathcal{C} \to \mathcal{I}$	$\mathcal{C} \to \mathcal{P}$	$\mathcal{P} \to \mathcal{C}$	Average
Source Only [9]	74.8 ± 0.3	83.9 ± 0.1	91.5 ± 0.3	78.0 ± 0.2	65.5 ± 0.3	91.2 ± 0.3	80.8
DANN [43]	82.0 ± 0.4	96.9 ± 0.2	99.1 ± 0.1	79.7 ± 0.4	68.2 ± 0.4	67.4 ± 0.5	82.2
MADA [11]	75.0 ± 0.3	87.9 ± 0.2	96.0 ± 0.3	88.8 ± 0.3	75.2 ± 0.2	92.2 ± 0.3	85.9
CDAN [13]	76.7 ± 0.3	90.6 ± 0.3	97.0 ± 0.4	90.5 ± 0.4	74.5 ± 0.3	93.5 ± 0.4	87.1
DAN [14]	74.5 ± 0.4	82.2 ± 0.2	92.8 ± 0.2	86.3 ± 0.4	69.2 ± 0.4	89.8 ± 0.4	82.4
RevGrad [15]	75.0 ± 0.6	86.0 ± 0.3	96.2 ± 0.4	87.0 ± 0.5	74.3 ± 0.5	91.5 ± 0.6	85.0
JAN [16]	76.8 ± 0.4	88.0 ± 0.2	94.7 ± 0.2	89.5 ± 0.3	74.2 ± 0.3	91.7 ± 0.3	85.7
ETD [49]	81.0	91.7	97.9	93.3	79.5	95.0	89.7
LDAuCID	**87.8** ± 1.4	**99.1** ± 0.2	**100** ± 0.0	**99.8** ± 0.0	**88.8** ± 1.0	**99.5** ± 0.3	**95.8**

Table 2: Classification accuracy for UDA tasks for ImageCLEF-DA dataset.

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	Average
Source Only [9]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DANN [43]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN [13]	49.0	69.3	74.5	**55.4**	66.0	**68.4**	**55.6**	**48.3**	**75.9**	**68.4**	**55.4**	**80.5**	**63.9**
DAN [14]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
JAN [16]	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
DJT [23]	39.7	50.4	62.4	39.5	54.3	53.1	36.7	39.2	63.5	52.2	45.4	70.4	50.6
LDAuCID	**48.3**	**67.4**	**74.1**	48.7	**61.9**	63.8	49.6	42.1	71.3	60.3	47.6	76.6	59.4

Table 3: Classification accuracy for UDA tasks of Office-Home dataset.

Method	A→C	A→D	A→W	W→A	W→D	W→C	D→A	D→W	D→C	C→A	C→W	C→D	Average
Source Only	84.6	81.1	75.6	79.8	98.3	79.6	84.6	96.8	80.5	92.4	84.2	87.7	85.4
DANN [43]	87.8	82.5	77.8	83.0	**100**	81.3	84.7	99.0	82.1	93.3	89.5	91.2	87.7
MMAN [51]	88.7	97.5	**96.6**	94.2	**100**	89.4	**94.3**	99.3	87.9	93.7	**98.3**	98.1	94.6
RevGrad [15]	85.7	89.2	90.8	93.8	98.7	86.9	90.6	98.3	83.7	92.8	88.1	87.9	88.9
DAN [14]	84.1	91.7	91.8	92.1	**100**	81.2	90.0	98.5	80.3	92.0	90.6	89.3	90.1
CORAL [52]	86.2	91.2	90.5	88.4	**100**	88.6	85.8	97.9	85.4	93.0	92.6	89.5	90.8
WDGRL [53]	87.0	93.7	89.5	93.7	**100**	89.4	91.7	97.9	90.2	93.5	91.6	94.7	92.7
LDAuCID	**99.6**	**100.0**	86.5	**96.1**	**100**	**99.8**	88.5	**100.0**	**95.7**	**99.3**	96.4	**99.8**	**96.8**

Table 4: Performance comparison for UDA tasks of Office-Caltech dataset.

Conclusions

- Tackling domain shift continually is a significant challenge in deep learning to preserve model generalizability after an initial training phase
- Maintaining the abstract knowledge about a domain that is learned as an internal distribution is a solution to both benefit from knowledge transfer to avoid leaning from scratch and to tackle catastrophic forgetting
- Determining the time to adapt the model automatically, i.e., the boundary between the domains, is the next step to extend our method
- Extensions to incremental learning setting need further exploration to make our proposed algorithm more practical

References

[1] Chen, Z. and Liu, B., 2018. Lifelong machine learning. Synthesis Lectures on Artificial Intelligence and Machine Learning, 12(3), pp.1-207.

[2] Rostami, M., Kolouri, S., Pilly, P. and McClelland, J., 2020, April. Generative continual concept learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 04, pp. 5545-5552).