Adapted Domain Specific Class Means

Gabriela Csurka Boris Chidlovskii Stéphane Clinchant

Xerox Research Centre Europe (XRCE)

firstname.lastname@xrce.xerox.com

Contribution

We assume that only class representatives are available from the sources.

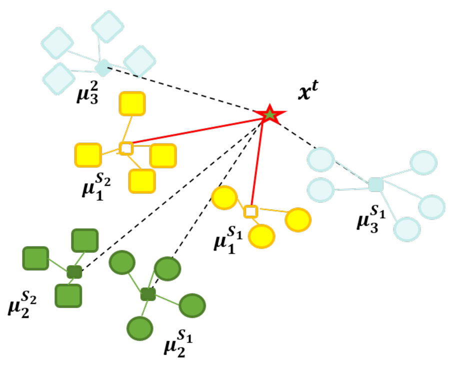
- Can be applied under the privacy concerns, with no access to all source instances (document images, medical field, etc).

Most domain adaptation (DA) methods needs to exploit the whole source dataset.

Main idea is to combine:

- Domain Specific Class Means (DSCM) classifier [4].
- Stacked Marginalized Denoising Autoencoders (sMDA) [2].

Domain Specific Class Means Classifier



Domain specific class means:

$$\mu_{\mathbf{d}}^{\mathbf{c}} = \frac{1}{\mathbf{N_d^c}} \sum_{\mathbf{x_i} \in \mathcal{D}_d \cap \mathcal{C}_c} \mathbf{x_i},$$

where N_d^c is the number of instances from class $\mathcal{C}_c$ in the domain. $\mathcal{D}_d$.

Target label prediction:

$$p(c|\mathbf{x}_i) = \frac{1}{Z_i} \sum_{d \in D} w_d e^{(-\frac{1}{2}\|\mathbf{x}_i - \mu_d^c\|)},$$

where:

- $Z_i = \sum_{c'} \sum_d w_d e^{(-\frac{1}{2}\|\mathbf{x}_i - \mu_d^{c'}\|)}$ is the normalizer,
- w_d are the domain weights (we use $w_d = 1$ for sources and 2 for the target).

Marginalized Denoising Autoencoder

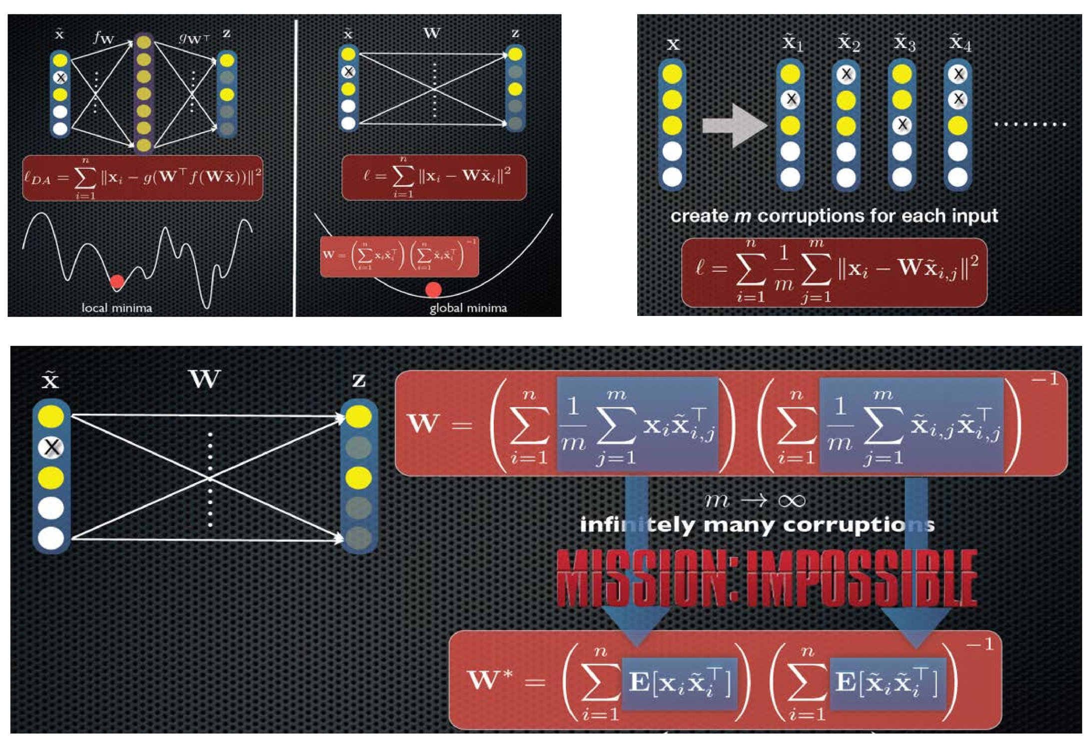


Image: Courtesy to M. Chen.

Main idea:

- Inputs $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ are corrupted m times by random feature removal (dropout) with the probability p (denoted by $\tilde{\boldsymbol{x}}_{ij}$ the j^{th} corrupted version of the input $\boldsymbol{x}_i$).
- Then reconstructed with a linear mapping $\mathbf{W}$ by minimizing the loss:

$$\mathcal{L}(\mathbf{W}) = \frac{\mathbf{1}}{\mathbf{2mn}} \sum_{\mathbf{j=1}}^{\mathbf{n}} \sum_{\mathbf{i=1}}^{\mathbf{m}} ||\mathbf{x_i} - \mathbf{W}\tilde{\mathbf{x}}_{\mathbf{i,j}}||^{\mathbf{2}}.$$

- If $m \to \infty$, the corruption can be marginalized out and $\mathbf{W}$ expressed in closed form as $\mathbf{W} = \mathbb{E}[\mathbf{P}]\,\mathbb{E}[\mathbf{Q}]^{-1}$, where:

$$\mathbb{E}[\mathbf{P}]_{ij} = \mathbf{S}_{ij} q_j \quad \text{and} \quad \mathbb{E}[\mathbf{Q}]_{ij} = \begin{bmatrix} \mathbf{S}_{ij} q_i q_j, & \text{if} & i \neq j \\ \mathbf{S}_{ij} q_i, & \text{if} & i = j \end{bmatrix} \quad \text{with :}$$

- $\mathbf{S} = \mathbf{X}\mathbf{X}^\top$ the covariance matrix of the uncorrupted data $\mathbf{X}$,
- $\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_m}]$ where $\mathbf{x}_i = [\boldsymbol{x}_i^\top, 1]^\top$ for the inputs $\boldsymbol{x}_i$,
- $q = [1-p, \ldots, 1-p, 1] \in R^{n+1}$, n is the feature dimension,
- and p is the noise level (by default 0.5).

Proposed Approach

- **sMDA framework to compute W for p from $\mathbf{X} = [\mathbf{X}_T, \mathbf{X}_S]$, where:**
  - $\mathbf{X}_S = [\mu_{\mathbf{s_1}}^{\mathbf{c_1}}, \ldots \mu_{\mathbf{s_S}}^{\mathbf{c_C}}]$, with $c_i \in C$ and $s_d \in D \setminus \{t_d\}$,
  - $\mathbf{X}_T = [\mathbf{x}_{t_1}, \ldots, \mathbf{x}_{t_n}]$ are the target instances.
- **Linear mapping followed by nonlinearities**:
  - the reconstructed class means $\hat{\mu}_{s_d}^{c_i} = f(\mathbf{W}\mu_{\mathbf{s_d}}^{\mathbf{c_i}})$,
  - the reconstructed unlabeled target examples $\hat{\mathbf{x}}_t = f(\mathbf{W}\mathbf{x}_t)$.
- **Stack l (we used 5) layers and then concatenate:**
  - "class means": $\breve{\mu}_{s_d}^{c_i} = [(\mu_{\mathbf{s_d}}^{\mathbf{c_i}})^\top, (\hat{\mu}_{\mathbf{c_i,s_d}}^{\mathbf{l}})^\top]^\top$, where $\hat{\mu}_{c_i,s_d}^{l} = f(\mathbf{W}_l\ \hat{\mu}_{c_i,s_d}^{l-1})$ and $\hat{\mu}_{c_i,s_d}^{0} = \mu_{\mathbf{s_d}}^{\mathbf{c_i}}$,
  - "targets": $\breve{\mathbf{x}}_t^l = [(\mathbf{x}_t)^\top, (\hat{\mathbf{x}}_t^l)^\top]^\top$, where $\hat{\mathbf{x}}_t^l = f(\mathbf{W}_l\ \hat{\mathbf{x}}_t^{l-1})$ and $\hat{\mathbf{x}}_t^0 = \mathbf{x}_t$.
- **Predict target labels**:

$$p(c|\mathbf{x}_t) = \frac{1}{\breve{\mathbf{Z}}_t} \sum_{d \in D} w_{s_d} e^{(-\frac{1}{2}\|\breve{\mathbf{x}}_t - \breve{\mu}_{s_d}^{c_i}\|)}, \quad \text{with :} \quad \breve{\mathbf{Z}}_t = \sum_{c'} \sum_{d} w_{s_d} e^{(-\frac{1}{2}\|\breve{\mathbf{x}}_t - \breve{\mu}_d^{c'}\|)}.$$

Experimental validation

Results on Office+Caltech10 [12] with SURF BOV:

- Semi-supervised scenario with the evaluation framework [8].
- ACM used with $p = 0.5, l = 5$ and $f = tanh$.

Method	C→A	D→A	W→A	A →C	D →C	W →C	A→D	C→D	W→D	A→W	C→W	D→W	Avg
ACM	52.4	50.1	50.3	37.3	37.5	36.8	58.9	59.7	63	68.1	69.6	75.2	55.3
GFK [8]	46.1	46.2	32.1	39.6	33.9	32.1	50.9	55	74.1	56.9	57	74.6	49.8
SA [5]	45.3	45.8	44.8	38.4	35.8	34.1	55.1	56.6	82.3	60.3	60.7	84.8	53.7
MMDT [9]	49.4	46.9	47.7	36.4	34.1	32.2	56.7	56.5	67	64.6	63.8	74.1	52.5
MLDSCM [4]	50.6	48.8	48.4	34.9	34.2	33.4	62.1	61.6	64.7	66.1	65.1	71.5	53.5
DIP-CC [1]	**61.8**	**56.9**	**53.4**	**47.8**	**44.2**	**43.6**	**67.5**	**65.8**	**92.6**	**72.5**	**69.9**	**89.1**	**63.7**

Results on Office 31 [8] using deep CNN network:

- Unsupervised scenario using the evaluation framework [8].
- ACM used with fc6 [10], $p = 0.5, l = 5$ and $f(u) = max(0, u)$ (RELU).

Method	A→W	D→W	W→D	Avg
ACM	67.5	92.9	94.1	84.8
SA [6]	47.2	91.8	92.4	77.1
CORAL [13]	48.4	**96.5**	**99.2**	81.4
DLID [3]	26.1	68.9	84.9	60
DDC [14]	59.4	92.5	91.7	81.2
DAN [11]	66	93.5	95.7	**85.1**
DAB [7]	67.3	94	93.7	85

Conclusion

We proposed a method that is:

- Simple and still powerful method with low computational cost.
- Requires only class means/prototypes from the source domains.
- Has only a few parameters (p, l, f) with reasonable default values.

Its advantages are:

- Can exploit multiple sources in both US and SS scenarios.
- Remains competitive with most existing methods.
- Can handle larger set of real scenarios (e.g. privacy issues).

References

[1] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *ICCV*, 2013.

[2] M. Chen, Z. Xu, K. Q. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. *ICML*, arXiv:1206.4683, 2012.

[3] S. Chopra, S. Balakrishnan, and R. Gopalan. DLID: Deep learning for domain adaptation by interpolating between domains. In *ICML*, 2013.

[4] G. Csurka, B. Chidlovskii, and F. Perronnin. Domain adaptation with a domain specific class means classifier. In *TASK-CV, ECCV workshop*, 2014.

[5] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

[6] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Subspace alignment for domain adaptation. *CoRR*, arXiv:1409.5241, 2014.

[7] Y. Ganin and V.S. Lempitsky. Unsupervised domain adaptation by backpropagation. *CORR*, 1409.7495, 2015.

[8] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[9] J. Hoffman, E. Rodner & J. Donhua & T. Darrell, and K. Saenko. Efficient learning of domain-invariant image representations. In *ICLR*, 2013.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[11] M. Long, Y. Cao, , J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[12] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[13] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. *CORR*, arXiv:1511.05547, 2015.

[14] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, arXiv:1412.3474, 2014.