(Oral) Visual Domain Adaptation with Manifold Embedded Distribution Alignment

Jindong Wang*, Wenjie Feng, Yiqiang Chen✉, Han Yu, Meiyu Huang, and Philip S. Yu
Inst. of Computing Tech., Chinese Academy of Sciences · NTU · China Academy of Space Tech. · UIC
jindongwang@outlook.com; Paper online: `https://arxiv.org/abs/1807.07258`; Code: `http://transferlearning.xyz`

1. Introduction

Visual domain adaptation aims to learn robust classifiers for the target domain by leveraging knowledge from a source domain. Existing methods are facing two significant challenges:

- *Degenerated feature transformation*: feature distortion often happens; subspace learning is not sufficient to reduce the distribution divergence.
- *Unevaluated distribution alignment*: existing methods fail to evaluate the different importance of marginal and conditional distributions.

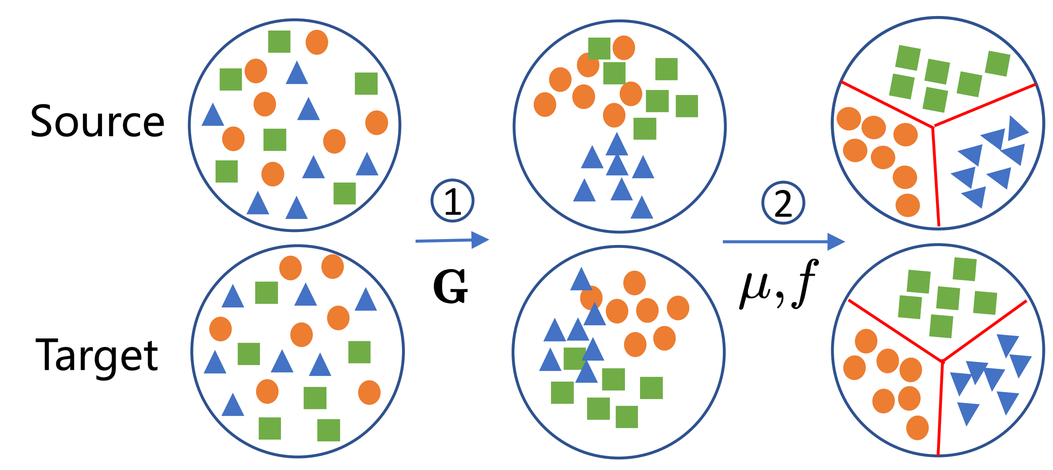
Our method: **Manifold Embedded Distribution Alignment (MEDA)**

$$f = \mathop{\arg\min}_{f \in \sum_{i=1}^{n} \mathcal{H}_K} \ell(f(g(\mathbf{x}_i)), y_i) + \eta ||f||_K^2 + \lambda \overline{D_f}(\mathcal{D}_s, \mathcal{D}_t) + \rho R_f(\mathcal{D}_s, \mathcal{D}_t)$$

MEDA learns a domain-invariant classifier in Grassmann manifold with structural risk minimization, while performing dynamic distribution alignment to **quantitatively** account for the relative importance of marginal and conditional distributions.

2. Our Method: MEDA

Manifold Embedded Distribution Alignment



- **G**: Manifold feature learning
- μ: Dynamic distribution alignment
- f: Prediction function
- R_f: Manifold regularization

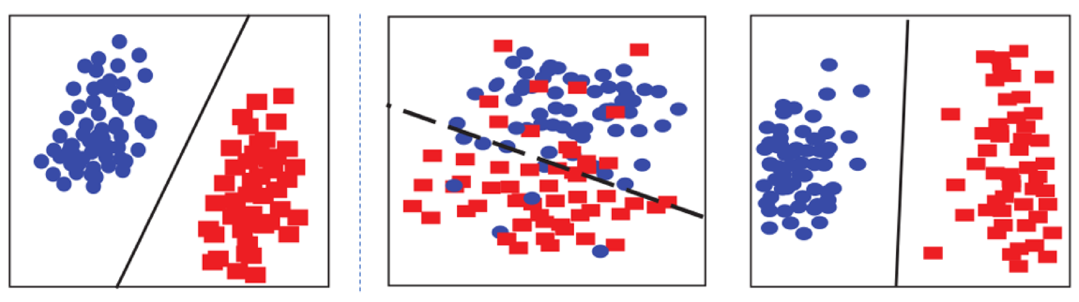
MEDA works in both traditional and deep frameworks.

2.1 Manifold Feature Learning

We adopted Geodesic Flow Kernel (GFK) [1] as the basic manifold learning method. GFK tries to model the domains with d-dimensional subspaces and then embed them into $\mathbb{G}$. Each original subspace can be seen as a point in $\mathbb{G}$. Therefore, the geodesic flow $\{\Phi(t) : 0 \le t \le 1\}$ between two points can draw a path for the two subspaces. If we let $\mathcal{S}_s = \Phi(0)$ and $\mathcal{S}_t = \Phi(1)$, then finding a geodesic flow from $\Phi(0)$ to $\Phi(1)$ equals to transforming the original features into an infinite-dimensional feature space.
The new features can be represented as $\mathbf{z} = g(\mathbf{x}) = \Phi(t)^T \mathbf{x}$. The inner product of transformed features $\mathbf{z}_i$ and $\mathbf{z}_j$ gives rise to a positive semidefinite geodesic flow kernel:

$$\langle \mathbf{z}_i, \mathbf{z}_j \rangle = \int_0^1 (\Phi(t)^T \mathbf{x}_i)^T (\Phi(t)^T \mathbf{x}_j)\, dt = \mathbf{x}_i^T \mathbf{G} \mathbf{x}_j$$

2.2 Dynamic Distribution Alignment



(a) Source (b) Target: Type I (c) Target: Type II

The **Dynamic Distribution Alignment (DDA)** is to tackle with the unevaluated distribution alignment challenge. The core is an **adaptive factor** to dynamically leverage the importance of marginal and conditional distributions. The DDA $\overline{D_f}$ is formed by linear combination of two distributions:

$$\overline{D_f}(\mathcal{D}_s, \mathcal{D}_t) = (1-\mu) D_f(P_s, P_t) + \mu \sum_{c=1}^{C} D_f^{(c)}(Q_s, Q_t)$$

where $\mu \in [0,1]$ is the adaptive factor and $c \in \{1, \cdots, C\}$ is the class indicator. $D_f(P_s, P_t)$ denotes the marginal distribution alignment, and $D_f^{(c)}(Q_s, Q_t)$ denotes the conditional distribution alignment for class c.
Taking the projected MMD, dynamic distribution alignment can be expressed as

$$\overline{D_f}(\mathcal{D}_s, \mathcal{D}_t) = (1-\mu) ||\mathbb{E}[f(\mathbf{z}_s)] - \mathbb{E}[f(\mathbf{z}_t)]||_{\mathcal{H}_K}^2 + \mu \sum_{c=1}^{C} ||\mathbb{E}[f(\mathbf{z}_s^{(c)})] - \mathbb{E}[f(\mathbf{z}_t^{(c)})]||_{\mathcal{H}_K}^2$$

The first quantitative calculation of μ:

$$\hat{\mu} \approx 1 - \frac{d_M}{d_M + \sum_{c=1}^{C} d_c}, d_M : \text{marginal } A\text{-distance}, d_c : \text{conditional } A\text{-distance [3]}$$

2.3 Learning

Using the representer theorem [2], f becomes

$$f(\mathbf{z}) = \sum_{i=1}^{n+m} \beta_i K(\mathbf{z}_i, \mathbf{z})$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots)^T \in \mathbb{R}^{(n+m)\times 1}$ is the coefficients vector and K is a kernel.
f can be reformulated as

$$f = \mathop{\arg\min}_{f \in \mathcal{H}_K} ||(\mathbf{Y} - \boldsymbol{\beta}^T \mathbf{K})\mathbf{A}||_F^2 + \eta \operatorname{tr}(\boldsymbol{\beta}^T \mathbf{K} \boldsymbol{\beta}) + \operatorname{tr}\left(\boldsymbol{\beta}^T \mathbf{K}(\lambda \mathbf{M} + \rho \mathbf{L}) \mathbf{K} \boldsymbol{\beta}\right)$$

Setting $\partial f / \partial \boldsymbol{\beta} = 0$, we obtain the solution

$$\boldsymbol{\beta}^\star = ((\mathbf{A} + \lambda \mathbf{M} + \rho \mathbf{L})\mathbf{K} + \eta \mathbf{I})^{-1} \mathbf{A} \mathbf{Y}^T$$

It shows that MEDA can **directly** learn the labels of the target domain, rather than train another classifier.

3. Experiments

On Office31, Office-Caltech10, USPS, MNIST, ImageNet, and VOC2007 datasets, MEDA shows:

- Over **3.5**% improvement in classification accuracy
- Over **11.6**% of error reduction
- Over **50.0**% drop of standard deviation

Method	A → D	A → W	D → A	D → W	W → A	W → D	Average
SVM	55.7	50.6	46.5	93.1	43.0	97.4	64.4
TCA	45.4	40.5	36.5	78.2	34.1	84.0	53.1
GFK	52.0	48.2	41.8	86.5	38.6	87.5	59.1
SA	46.2	42.5	39.3	78.9	36.3	80.6	54.0
DANN	34.0	34.1	20.1	62.0	21.2	64.4	39.3
CORAL	57.1	53.1	51.1	94.6	47.3	98.2	66.9
AlexNet	63.8	61.6	51.1	95.4	49.8	99.0	70.1
DDC	64.4	61.8	52.1	95.0	52.2	98.5	70.6
DAN	67.0	68.5	54.0	96.0	53.1	99.0	72.9
RTN	71.0	**73.3**	50.5	**96.8**	51.0	99.6	73.7
DCORAL	66.4	66.8	52.8	95.7	51.5	99.2	72.1
DUCDA	68.3	68.3	53.6	96.2	51.6	**99.7**	73.0
MEDA	**69.5**	69.9	**58.0**	94.0	**56.0**	96.8	**74.0**

With regards to the different importance of marginal and conditional distributions:

- There **does** exist different importance between these two distributions (μ)
- MEDA can provide an **accurate**, even **better** calculation of μ compared to gridsearch
- DDA can also be included in the **deep** transfer learning frameworks

Task	C → A	W → D	C → A (DeCaf)	W → C (DeCaf)	M → U	I → V
μ_{opt}	57.0	89.2	93.4	88.0	89.4	67.6
$\hat{\mu}$	56.5	88.5	93.4	93.2	89.5	67.3
Performance Variation	-0.9%	-0.8%	0	**+5.9%**	**+0.1%**	-0.4%

References

[1] B. Gong, et al. Geodesic flow kernel for unsupervised domain adaptation. CVPR 2012.
[2] M. Belkin, et al. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. JMLR 2006.
[3] S. Ben-David, et al. Analysis of Representations for Domain Adaptation. NIPS 2007.