Mind the class weight bias: weighted maximum mean discrepancy for unsupervised domain adaptation

Hongliang Yan¹, Yukang Ding¹, Peihua Li², Qilong Wang², Yong Xu³, Wangmeng Zuo¹,*
Domain Adaptation

Problem:

Training and test sets are related but under different distributions.

Methodology:

- Learn feature space that combine discriminativeness and domain invariance.

\[
\text{minimize} \quad \text{source error} + \text{domain discrepancy}
\]

Maximum Mean Discrepancy (MMD)

• representing distances between distributions as distances between mean embeddings of features

\[
\text{MMD}^2(s, t) = \sup_{\|\phi\|_H \leq 1} \| E_{x \sim s} [\phi(x)] - E_{x' \sim t} [\phi(x')] \|^2_H
\]

• An empirical estimate

\[
\text{MMD}^2(D_s, D_t) = \left\| \frac{1}{M} \sum_{i=1}^{M} \phi(x^s_i) - \frac{1}{N} \sum_{j=1}^{t} \phi(x^t_j) \right\|^2_H
\]
Motivation

- *Class weight bias* cross domains remains unsolved but ubiquitous

\[
\text{MMD}^2(D_s, D_t) = \| \frac{1}{M} \sum_{i=1}^{M} \phi(x_i^s) - \frac{1}{N} \sum_{j=1}^{T} \phi(x_j^t) \|_H^2
\]
Motivation

• *Class weight bias* cross domains remains unsolved but ubiquitous

\[
\text{MMD}^2(D_s, D_t) = \| \frac{1}{M} \sum_{i=1}^{M} \phi(x^s_i) - \frac{1}{N} \sum_{j=1}^{N} \phi(x^t_j) \|_H^2
\]

\[
\| \sum_{c=1}^{C} w_c E_c(\phi(x^s)) - \sum_{c=1}^{C} w'_c E_c(\phi(x^t)) \|_H^2
\]

\[
w^s_c = \frac{M_c}{M} \text{ and } w'_c = \frac{N_c}{N}
\]
Motivation

• **Class weight bias** cross domains remains unsolved but ubiquitous

\[
\text{MMD}^2(D_s,D_t) = \left\| \frac{1}{M} \sum_{i=1}^{M} \phi(x_i^s) - \frac{1}{N} \sum_{j=1}^{N} \phi(x_j^t) \right\|_H^2
\]

**Effect of class weight bias should be removed:**

1. Changes in sample selection criteria

\[
\left\| \sum_{c=1}^{C} w_c^s E_c(\phi(x^s)) - \sum_{c=1}^{C} w_c^t E_c(\phi(x^t)) \right\|_H^2
\]

\[w_c^s = M_c / M \quad \text{and} \quad w_c^t = N_c / N\]
Motivation

• **Class weight bias** cross domains remains unsolved but ubiquitous

\[ \text{MMD}^2(D_s, D_t) = \| \frac{1}{M} \sum_{i=1}^{M} \phi(x_i^s) - \frac{1}{N} \sum_{j=1}^{N} \phi(x_j^t) \|_H^2 \]

**Effect of class weight bias should be removed:**

1. Changes in sample selection criteria

\[ \| \sum_{c=1}^{C} w_c^s E_c(\phi(x^s)) - \sum_{c=1}^{C} w_c^t E_c(\phi(x^t)) \| \]

\[ w_c^s = M_c / M \quad \text{and} \quad w_c^t = N_c / N \]

Figure 2. Class prior distribution of three digit recognition datasets.
Motivation

- *Class weight bias* cross domains remains unsolved but ubiquitous

\[ \text{MMD}^2(D_s, D_t) = \left\| \frac{1}{M} \sum_{i=1}^{M} \phi(x_i^s) - \frac{1}{N} \sum_{j=1}^{N} \phi(x_j^t) \right\|_H^2 \]

**Effect of class weight bias should be removed:**

1. Changes in sample selection criteria
2. Applications are not concerned with class prior distribution

\[ \left\| \sum_{c=1}^{C} w_c^s E_c(\phi(x^s)) - \sum_{c=1}^{C} w_c^t E_c(\phi(x^t)) \right\|_H^2 \]

\[ w_c^s = \frac{M_c}{M} \text{ and } w_c^t = \frac{N_c}{N} \]
Motivation

- **Class weight bias** cross domains remains unsolved but ubiquitous

\[
\text{MMD}^2(D_s, D_t) = \| \frac{1}{M} \sum_{i=1}^{M} \phi(x^s_i) - \frac{1}{N} \sum_{j=1}^{N} \phi(x^t_j) \|_H^2
\]

\[
\| \sum_{c=1}^{C} w^s_c E_c(\phi(x^s)) - \sum_{c=1}^{C} w^t_c E_c(\phi(x^t)) \|_H^2
\]

- **Effect of class weight bias should be removed:**
  1. Changes in sample selection criteria
  2. Applications are not concerned with class prior distribution

MMD can be minimized by either learning domain invariant representation or preserving the class weights in source domain.
Weighted MMD

Main idea: reweighting classes in source domain so that they have the same class weights as target domain

- Introducing an auxiliary weight $\alpha_c$ for each class $c$ in source domain

\[
MMD^2(D_s, D_t) = \| \frac{1}{M} \sum_{i=1}^{M} \phi(x^s_i) - \frac{1}{N} \sum_{j=1}^{N} \phi(x^t_j) \|^2_{H}
\]

\[
\| \sum_{c=1}^{C} w^s_c E_c(\phi(x^s)) - \sum_{c=1}^{C} w^t_c E_c(\phi(x^t)) \|^2_{H}
\]

\[
\alpha_c = \frac{w^t_c}{w^s_c}
\]
Weighted MMD

Main idea: reweighting classes in source domain so that they have the same class weights as target domain

• Introducing an auxiliary weight $\alpha_c$ for each class $c$ in source domain

$$\text{MMD}^2(D_s, D_t) = \frac{1}{M} \sum_{i=1}^{M} \phi(x_i^s) - \frac{1}{N} \sum_{j=1}^{t} \phi(x_j^t) \|_H^2$$

$$\text{MMD}_w^2(D_s, D_t) = \frac{1}{M} \sum_{i=1}^{M} \alpha_{y_i^s} \phi(x_i^s) - \frac{1}{N} \sum_{j=1}^{t} \phi(x_j^t) \|_H^2$$

$\alpha_c = \frac{w_c^t}{w_c^s}$

$$\| \sum_{c=1}^{C} w_c^s E_c (\phi(x^s)) - \sum_{c=1}^{C} w_c^t E_c (\phi(x^t)) \|_H^2$$

$$\| \sum_{c=1}^{C} w_c^t E_c (\phi(x^s)) - \sum_{c=1}^{C} w_c^s E_c (\phi(x^t)) \|_H^2$$
Weighted DAN

1. Replace MMD with weighted MMD item in DAN[4]:

\[
\min_w \frac{1}{M} \sum_{i=1}^{M} \ell(x_i^s, y_i^s; W) + \lambda \sum_{l \in \{l_1, \ldots, l_L\}} \text{MMD}_l(D_s^l, D_t^l)
\]

Weighted DAN

1. Replace MMD with weighted MMD item in DAN[4]:

$$\min_{w, \alpha} \frac{1}{M} \sum_{i=1}^{M} \ell(x_i^s, y_i^s; W) + \lambda \sum_{l \in \{l_1, \ldots, l_L\}} \text{MMD}_{l,w}(D^l_s, D^l_t)$$

Weighted DAN

1. Replace MMD with Weighted MMD item in DAN[4]:

$$\min_{\mathbf{w}} \frac{1}{M} \sum_{i=1}^{M} \ell(x_i^s, y_i^s; \mathbf{W}) + \lambda \sum_{l \in \{l_1, \ldots, l_L\}} \text{MMD}_l(D_s^l, D_t^l)$$

2. To further exploit the unlabeled data in target domain, empirical risk is considered as semi-supervised model in [5]:

$$\min_{\mathbf{w}, (\hat{y}_i)_{i=1}^{M, \alpha}} \frac{1}{M} \sum_{i=1}^{M} \ell(x_i^s, y_i^s; \mathbf{W}) + \gamma \frac{1}{N} \sum_{j=1}^{N} \ell(x'_i, \hat{y}_i'; \mathbf{W}) + \lambda \sum_{l \in \{l_1, \ldots, l_L\}} \text{MMD}_{l,w}(D_s^l, D_t^l)$$

Optimization: an extension of CEM[6]

Parameters to be estimated including three parts, i.e., $W, \alpha, \{\hat{y}_j\}_{j=1}^N$

The model is optimized by alternating between three steps:

• E-step:
  Fixed $W$, estimating the class posterior probability $p(y_j^i = c \mid x_j^i)$ of target samples:
  
  $p(y_j^i = c \mid x_j^i) = g(x_j^i, W)$

Optimization: an extension of CEM[6]

Parameters to be estimated including three parts, i.e., $W, \alpha, \{\hat{y}_j^t\}_{j=1}^N$

The model is optimized by alternating between three steps:

• **E-step:**
  Fixed $W$, estimating the class posterior probability $p(y_j^t = c \mid x_j^t)$ of target samples:
  $$p(y_j^t = c \mid x_j^t) = g(x_j^t, W)$$

• **C-step:**
  ① Assign the pseudo labels $\{\hat{y}_j^t\}_{j=1}^N$ on target domain: $\hat{y}_j^t = \arg \max_c p(y_j^t = c \mid x_j^t)$
  ② update the auxiliary class-specific weights $\alpha$ for source domain:
  $$\alpha_c = \hat{w}_c^t / w_c^s$$
  where $\hat{w}_c^t = \sum_j 1_c(\hat{y}_j^t) / N$
  $1_c(x)$ is an indictor function which equals 1 if $x = c$, and equals 0 otherwise.

Optimization: an extension of CEM[6]

Parameters to be estimated including three parts, i.e., $W, \alpha, \{\hat{y}_j^i\}_{j=1}^N$

The model is optimized by alternating between three steps:

• **M-step:**

  Fixed $\{\hat{y}_j^i\}_{j=1}^N$ and $\alpha$, updating $W$. The problem is reformulated as:

  $$
  \min_W \frac{1}{M} \sum_{i=1}^M \ell(x_i^s, y_i^s; W) + \gamma \sum_{j=1}^N \ell(x_i^l, y_i^l; W) + \lambda \sum_{l \in \{l_1, \ldots, l_L\}} \text{MMD}_{l,w}(D_s^l, D_l^l)
  $$

  The gradient of the three items is computable and $W$ can be optimized by using a mini-batch SGD.

Experimental results

• Comparison with state-of-the-arts

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet [21]</td>
<td>84.0±0.3</td>
<td>77.9±0.4</td>
<td>81.0±0.4</td>
<td>91.3±0.2</td>
<td>83.2±0.3</td>
<td>89.1±0.2</td>
<td>84.0</td>
</tr>
<tr>
<td>LapCNN (AlexNet) [38]</td>
<td>83.6±0.6</td>
<td>77.8±0.5</td>
<td>80.6±0.4</td>
<td>92.1±0.3</td>
<td>81.6±0.4</td>
<td>87.8±0.4</td>
<td>83.9</td>
</tr>
<tr>
<td>DDC (AlexNet) [36]</td>
<td>84.3±0.5</td>
<td>76.9±0.4</td>
<td>80.5±0.2</td>
<td>91.3±0.3</td>
<td>85.5±0.3</td>
<td>89.1±0.3</td>
<td>84.6</td>
</tr>
<tr>
<td>DAN (AlexNet) [25]</td>
<td>86.0±0.5</td>
<td>81.5±0.3</td>
<td>82.0±0.4</td>
<td>92.0±0.3</td>
<td>92.6±0.4</td>
<td>90.5±0.2</td>
<td>87.3</td>
</tr>
<tr>
<td>WDAN (AlexNet)</td>
<td><strong>86.9±0.1</strong></td>
<td><strong>84.1±0.2</strong></td>
<td><strong>83.9±0.1</strong></td>
<td><strong>93.1±0.2</strong></td>
<td><strong>93.6±0.2</strong></td>
<td><strong>93.4±0.2</strong></td>
<td><strong>89.2</strong></td>
</tr>
<tr>
<td>WDAN* (AlexNet)</td>
<td>87.1±0.2</td>
<td>85.1±0.3</td>
<td>85.2±0.2</td>
<td>93.2±0.1</td>
<td>93.5±0.3</td>
<td>94.5±0.2</td>
<td>89.8</td>
</tr>
<tr>
<td>GoogLeNet [34]</td>
<td>91.3±0.2</td>
<td>88.2±0.3</td>
<td>88.9±0.3</td>
<td>95.2±0.1</td>
<td>92.5±0.2</td>
<td>94.7±0.3</td>
<td>91.8</td>
</tr>
<tr>
<td>DDC (GoogLeNet) [36]</td>
<td>91.4±0.2</td>
<td>88.7±0.3</td>
<td>89.0±0.4</td>
<td>95.3±0.2</td>
<td>93.0±0.1</td>
<td>94.9±0.4</td>
<td>92.1</td>
</tr>
<tr>
<td>DAN (GoogLeNet) [25]</td>
<td>91.4±0.3</td>
<td>89.7±0.2</td>
<td>89.1±0.4</td>
<td>95.5±0.2</td>
<td>93.1±0.3</td>
<td>95.3±0.1</td>
<td>92.3</td>
</tr>
<tr>
<td>WDAN (GoogLeNet)</td>
<td><strong>92.2±0.2</strong></td>
<td><strong>91.0±0.5</strong></td>
<td><strong>89.8±0.3</strong></td>
<td><strong>95.5±0.3</strong></td>
<td><strong>95.4±0.2</strong></td>
<td><strong>95.5±0.5</strong></td>
<td><strong>93.2</strong></td>
</tr>
<tr>
<td>VGGnet-16 [32]</td>
<td>89.6±0.4</td>
<td>88.1±0.4</td>
<td>85.4±0.5</td>
<td>93.7±0.2</td>
<td>94.3±0.2</td>
<td>93.7±0.2</td>
<td>90.8</td>
</tr>
<tr>
<td>DAN (VGGnet-16) [25]</td>
<td>91.2±0.2</td>
<td>90.6±0.3</td>
<td>87.1±0.4</td>
<td>95.7±0.2</td>
<td>95.3±0.3</td>
<td>94.7±0.1</td>
<td>92.4</td>
</tr>
<tr>
<td>WDAN (VGGnet-16)</td>
<td><strong>91.4±0.2</strong></td>
<td><strong>91.0±0.2</strong></td>
<td><strong>89.0±0.3</strong></td>
<td><strong>95.7±0.1</strong></td>
<td><strong>95.8±0.2</strong></td>
<td><strong>95.9±0.3</strong></td>
<td><strong>93.1</strong></td>
</tr>
</tbody>
</table>

Table 1. Experimental results on office-10+Caltech-10
Experimental results

• Empirical analysis

Figure 3. Performance of various model under different class weight bias.

Figure 4. Visualization of the learned features of DAN and weighted DAN.
Summary

• Introduce class-specific weight into MMD to reduce the effect of class weight bias cross domains.

• Develop WDAN model and optimize it in an CEM framework.

• Weighted MMD can be applied to other scenarios where MMD is used for distribution distance measurement, e.g., image generation.
Thanks!

Paper & code are available
Code: https://github.com/yhldhit/WMMD-Caffe