Reshaping Visual Datasets for Domain Adaptation

Review by: David Carlson

Boqing Gong, Kristen Grauman, and Fei Sha
Motivation and Contributions

- Many learning algorithms assume that the *domain* of the training set and the testing set are the same.
- In real-world applications, the distributions are often mismatched.
- Propose novel learning methods to automatically reshape datasets into domains that possess:
  - *Maximum distinctiveness* to identify domains that are maximally different in distribution from each other.
  - *Maximum learnability* to identify domains from which we can derive strong discriminative models.
Data instances are in the form of \((x_m, y_m)\) where \(x_m \in \mathbb{R}^D\) is the feature vector and \(y_m \in \{1, \ldots, C\}\) is the corresponding label out of \(C\) categories.

Assume each data instance comes from a latent domain \(z_m \in 1, \ldots, K\).
The domain is estimated by an empirical distribution

\[ \hat{P}_k(x) = \frac{1}{M_k} \sum_m \delta_{x_m} z_{mk} \]  

(1)

with \( z_{mk} = 1 \) if sample \( m \) is in domain \( k \) and \( M_k \) the number of samples in domain \( k \).
Maximally distinctive distributions

- Use a kernel-based method to measure distance between distributions
- Compute distance in the reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ induced by the kernel function,

\[
d(k, k') = \left\| \frac{1}{M_k} \sum_m K(\cdot, x_m)z_{mk} - \frac{1}{M'_{k'}} \sum_m K(\cdot, x_m)z_{mk'} \right\|_{\mathcal{H}}^2
\]

- Total domain distinctiveness (TDD) is defined as:

\[
TDD(K) = \sum_{k \neq k'} d(k, k')
\]  

(2)

- Details on calculation later
We have several constraints on assigning points to domains:

- Each sample can only belong to a single domain:
  \[ \sum_k z_{mk} = 1, \quad z_{mk} \in \{0, 1\} \]  \hspace{1cm} (3)

- Also, a *label prior constraint* (LPC) enforcing that the class labels in each domain are identical in each domain:
  \[ \frac{1}{M_k} \sum_{m=1}^{M} z_{mk} y_{mc} = \frac{1}{M} \sum_{m=1}^{M} y_{mc}, \quad y_{mc} = 1 \text{ if } y_m = c \]  \hspace{1cm} (4)
Relaxation

- Previous constraints are difficult to fit
- Relax the model with $\beta_{mk} = z_{mk}/M_k$ and let $\beta_k$ lie on the simplex $\Delta$
- Relaxed optimization problem with $K$ the $M \times M$ kernel matrix:

\[
\max_\beta \sum_{k \neq k'} \text{TDD}(K) = \sum_{k \neq k'} (\beta_k - \beta_{k'})^T K (\beta_k - \beta_{k'})
\]

s.t.

\[
\frac{1}{M} \leq \sum_k \beta_{mk} \leq \frac{1}{C (1 - \delta) / M}
\]

\[
(1 - \delta) / M \sum_m y_{mc} \leq \sum_m \beta_{mk} y_{mc} \leq (1 + \delta) / M \sum_m y_{mc}
\]

- Assign $x_m$ to the domain with the highest $\beta_{mk}$
Maximally learnable domains

▶ How many domains $K$ are there?
▶ Propose domain-wise cross-validation (DWCV) to identify the optimal $K$
▶ Given domain assignments, build a discriminative classifier, denote the cross-validation accuracy for the $k^{th}$ domain by $A_k$, and

$$A(K) = \frac{1}{M} \sum_{k=1}^{K} M_k A_k$$

▶ Choose $K^* = \arg \max_K A(K)$ for $2 \leq K^* \leq \min\{M/(NC), C\}$, $N$ is the number of folds
Experimental Results

- For object recognition, images are from Caltech-256 (C), Amazon (A), DSLR (D), and Webcam (W), which share 10 categories among the 4 datasets
  - Images represented with bag-of-visual-words by extracting SURF features from the images and using K-means to build a codebook of 800 clusters
- For action recognition, data is from IXMAS multi-view action dataset of 5 views of 11 actions
  - Shape-form descriptors are used as features
Look at upper bound on classification accuracy. \( S \) are the source data sets and \( T \) are the target data sets. \( G_{\text{ORIG}} \) each given dataset as a given domain, \( G_{\text{OTHER}} \) is a competing method, and \( G_{\text{OURS}} \) learns the domains from the data.

**Table 1: Oracle recognition accuracy on target domains by adapting original or identified domains**

<table>
<thead>
<tr>
<th>( S )</th>
<th>A, C</th>
<th>D, W</th>
<th>C, D, W</th>
<th>Cam 0, 1</th>
<th>Cam 2, 3, 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>D, W</td>
<td>A, C</td>
<td>A</td>
<td>Cam 2, 3, 4</td>
<td>Cam 0, 1</td>
</tr>
<tr>
<td>( G_{\text{ORIG}} )</td>
<td>41.0</td>
<td>32.6</td>
<td>41.8</td>
<td>44.6</td>
<td>47.1</td>
</tr>
<tr>
<td>( G_{\text{OTHER}} ) [20]</td>
<td>39.5</td>
<td>33.7</td>
<td>34.6</td>
<td>43.9</td>
<td>45.1</td>
</tr>
<tr>
<td>( G_{\text{OURS}} )</td>
<td>42.6</td>
<td>35.5</td>
<td>44.6</td>
<td>47.3</td>
<td>50.3</td>
</tr>
</tbody>
</table>
Datasets in practice are not labeled with their corresponding training domain

- UNION: The most naive way is to combine all the source domains into a single dataset and adapt from this mega domain to the target domains. We use this as a baseline.
- ENSEMBLE: A more sophisticated strategy is to adapt each source domain to the target domain and combine the adaptation results by combining multiple classifiers.
- MATCHING This strategy compares the empirical (marginal) distribution of the source domains and the target domains and selects the single source domain that has the smallest difference to the target domain to adapt.

Table 2: Adaptation recognition accuracies, using original and identified domains with different multi-source adaptation methods

<table>
<thead>
<tr>
<th>Latent Domains</th>
<th>Multi-DA method</th>
<th>A, C</th>
<th>D, W</th>
<th>C, D, W</th>
<th>Cam 0, 1</th>
<th>Cam 2, 3, 4</th>
<th>Cam 2, 3, 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORIGINAL [20]</td>
<td>UNION</td>
<td>41.7</td>
<td>35.8</td>
<td>41.0</td>
<td>45.1</td>
<td>47.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ENSEMBLE</td>
<td>31.7</td>
<td>34.4</td>
<td>38.9</td>
<td>43.3</td>
<td>29.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MATCHING</td>
<td>39.6</td>
<td>34.0</td>
<td>34.6</td>
<td>43.2</td>
<td>45.2</td>
<td></td>
</tr>
<tr>
<td>OURS</td>
<td>ENSEMBLE</td>
<td>38.7</td>
<td>35.8</td>
<td>42.8</td>
<td>45.0</td>
<td>40.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MATCHING</td>
<td>42.6</td>
<td>35.5</td>
<td>44.6</td>
<td>47.3</td>
<td>50.3</td>
<td></td>
</tr>
</tbody>
</table>
Let 5 views of action recognition be labeled $A, B, C, D, E$, and $F = \{D, E\}$

Training used $A, B, C$ and identified domains $A', B', C'$

From identified domain only uses one of the identified domains

Conditional reshaping learns testing domains based on the identified training domains

Table 3: Results of reshaping the test set when it consists of data from multiple domains.

<table>
<thead>
<tr>
<th></th>
<th>From identified (Reshaping training only)</th>
<th>No reshaping</th>
<th>Conditional reshaping</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$A' \to F$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cam 012</td>
<td>36.4</td>
<td>37.3</td>
<td>38.5</td>
</tr>
<tr>
<td>Cam 123</td>
<td>40.4</td>
<td>39.9</td>
<td>41.1</td>
</tr>
<tr>
<td>Cam 234</td>
<td>46.5</td>
<td>47.8</td>
<td>49.2</td>
</tr>
<tr>
<td>Cam 340</td>
<td>50.7</td>
<td>52.3</td>
<td>54.9</td>
</tr>
<tr>
<td>Cam 401</td>
<td>43.6</td>
<td>43.3</td>
<td>44.8</td>
</tr>
</tbody>
</table>
Figure 1: Exemplar images from the original and identified domains after reshaping. Note that identified domains contain images from both datasets.
Figure 1: Exemplar images from the original and identified domains after reshaping. Note that identified domains contain images from both datasets.

Figure 2: Domain-wise cross-validation (DWCV) for choosing the number of domains.