Domain Adaptation for Visual Applications

Gabriela Csurka

Xerox Research Centre Europe

6 Rue de Maupertuis, 38240, Meylan, France

Gabriela.Csurka@xrce.xerox.com
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Domain Adaptation (DA)

Leveraging labeled data in one or more related domains, referred to as source domains, to learn a classifier for unseen data in a target domain.

Sources

Target

- Unsupervised (US) DA when no label is available for the target
- Semi-supervised (SS) DA when we have a few labels in the target

Image: Courtesy to S.J. Pan

G. Csurka, DA for Visual Applications
Particular case of the transductive transfer learning where

- domains are different but the task is the same (e.g. same classes)
Example scenarios

Object recognition

Object detection

Document image categorization

Sentiment analyses

Action recognition

Speech recognition

G. Csurka, DA for Visual Applications
Why not just use source model?

- **Source Domain**
  - European Parliament Text
- **Target Domain**
  - Research Articles in Social Sciences.
- **No Adaptation**
  - 28% Mean Average Precision (MAP)
- **With Adaptation** 34%

- **German query**:
  - Selbstmord von Jugendlichen
- **Baseline translation**:
  - suicide of young (MAP = 0.3210)
- **Adaptation**:
  - suicide of adolescents (MAP = 0.5277)

**Statistical Machine Translation**

**Sentiment analyses**

**Object recognition**

Applying the models learned on the source directly often performs poorly!
Domain shift/distribution mismatch

Underlying causes:

▶ Image categorization
  • different point of views, acquisition time and conditions,
▶ Audio recognition
  • different persons, environment, recording quality
▶ Document image categorization
  • differences in appearance, layout
▶ Activity recognition
  • different persons, environment, context
▶ Semantic analyses
  • different topics, vocabularies, ...
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Amazon review dataset\(^1\) (AMT)

Products reviews in different domains
- kitchen (K), dvd (D), books (B) and electronics (E)
- 2 classes, about 5,000 document for each class
- TFIDF from processed text

\(^1\)Blitzer et al., Domain adaptation with coupled subspaces, AIS11

G. Csurka, DA for Visual Applications
Object recognition:

- Amazon (A), Caltech (C), Dslr (D), Webcam (W)
- 3 domains and 31 classes in OFF31 and 4 domains and 10 classes in OC10
- SURF BOV and Decaf6 (CNN activation) features
  - using all source examples (a)$\varepsilon$e
  - using several subset of the source data and average(s)$\varepsilon$e

---

$^2$Saenko et al., Adapting visual category models to new domains, ECCV10
$^3$Gong et al., Reshaping visual datasets for domain adaptation, NIPS13
The ImageCLEF’14 DA Challenge\(^4\) (ICDA)

Object recognition:
- Caltech (C), ImageNet (I), Pascal (P), Bing (B), SUN (S)
- 12 classes, about 60 documents for each class
- SIFT BOV features from images

\(^4\) http://www.imageclef.org/2014/adaptation

G. Csurka, DA for Visual Applications
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Different solutions

Correcting sampling bias

\[ x \rightarrow z, \quad \text{s.t.} \]

\[ P_S(z, y) \approx P_T(z, y) \]

Adjusting mismatched models

Image: Courtesy to Boqing Gong.

G. Csurka, DA for Visual Applications
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives
Learn a classifier such that

- more weights are put on the examples that are similar to target instances
- less weights (or removing) on those that are less similar
How to estimate the weights

- Using the classifier that distinguishes between source and target examples (Bickel et al. ICML'07)

\[ \alpha(x_i^s) = \frac{1}{p(y_i^s = s|x_i^s, \theta)} \]

- Considering the ratio between the densities estimated for source and target domains (Sugiyama et al. NIPS’07, Kanamori et al. JMLR’09)

\[ \alpha(x) = \frac{P_T(x)}{P_S(x)} \approx \sum_l \alpha_l \phi_l(x) \]
Maximum Mean Discrepancy

Minimizing the Maximum Mean Discrepancy (Huang et al. NIPS’06)

\[ MMD(S, T) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} \phi(x^s_i) - \frac{1}{N_t} \sum_{j=1}^{N_t} \phi(x^t_j) \right\|_\mathcal{H} \]

where \( \mathcal{H} \) is the RKHS (reproducing kernel Hilbert space) associated with the kernel \( k \), and \( \phi(x) = \langle k(x), . \rangle \).

Empirically:

\[ MMD(S, T) = \left[ \frac{1}{N^2_s} \sum_{i,j=1}^{N_s} k(x^s_i, x^s_j) - 2 \frac{1}{N_s N_t} \sum_{i,j=1}^{N_s N_t} k(x^s_i, x^t_j) + \frac{1}{N^2_t} \sum_{j,j=1}^{N_t} k(x^t_i, x^t_j) \right] \]

with \( k \) being e.g. the Gaussian Kernel.
Transfer Adaptive Boosting\(^5\)

- **Hedge (β)** [Freund et al. 1997]
  - To decrease the weights of the misclassified data

- **AdaBoost** [Freund et al. 1997]
  - To increase the weights of the misclassified data

\[\text{Source domain labeled data} \rightarrow \text{Hedge (β)} \rightarrow \text{AdaBoost} \rightarrow \text{Target domain unlabeled data} \]

- **Classifiers trained on re-weighted labeled data**

- **The whole training data set**

Image: Courtesy to S.J. Pan.

---

\(^5\) Dai et al., Boosting for transfer learning, ICML'07.

G. Csurka, DA for Visual Applications
TrAdaboost result

- 20 Newsgroups - text categorization across newsgroups.
- Abalone - seven physical measurements of abalone sea snails.
- Wine - red wine physical and chemical characteristics versus white wine.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AdaBoost</th>
<th>TrAdaBoost</th>
<th>Fixed-Cost (1.1, 1.2, 1.3)</th>
<th>Dynamic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sci vs Talk</td>
<td>0.552</td>
<td>0.577</td>
<td>0.581</td>
<td>0.618</td>
</tr>
<tr>
<td>Rec vs Sci</td>
<td>0.546</td>
<td>0.572</td>
<td>0.588</td>
<td>0.631</td>
</tr>
<tr>
<td>Rec vs Talk</td>
<td>0.585</td>
<td>0.660</td>
<td>0.670</td>
<td>0.709</td>
</tr>
<tr>
<td>Wine Quality</td>
<td>0.586</td>
<td>0.604</td>
<td>0.605</td>
<td>0.638</td>
</tr>
<tr>
<td>Abalone Age</td>
<td>0.649</td>
<td>0.689</td>
<td>0.682</td>
<td>0.740</td>
</tr>
</tbody>
</table>

- **TrAdaBoost** - Transfer Adaptive Boosting, Dai et al., ICML'07.
- **Dynamic** - Dynamic updates for TrAdaBoost, Al-Stouhi and Reddy, PKDD’11.
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features
5. Multiple sources
6. Deep Learning
7. Conclusion and Perspectives
Adaptive SVM\(^6\) (A-SVM)

The target classifier:

\[
f^t(x) = \sum_{k=1}^{M} \beta_k f_k^a(x) + \sum_{i=1}^{N} \hat{\alpha}_i y_i K(x, x_i)
\]

leverages multiple auxiliary classifiers \(f_k^a\).

---

\(^6\) Yang et al., Cross-domain video concept detection using adaptive SVMs, MM’07
Domain Adaptation SVM\(^7\) (DASVM)

Adapt iteratively (until stopping criteria reached) the classifier built with the source:

- add to negatives the first k predicted target \(h(x_t) > 0\) with highest margin
- add to positives the first k predicted target \(h(x_t) < 0\) with highest margin
- remove the first k positive and k negative source instances with highest margin

\(^7\)Bruzzone and Marconcini, Domain Adaptation Problems: A DASVM Classification Technique and a Circular Validation Strategy PAMI’10.
Adaptive MKL\textsuperscript{8}

- Proposed formulation of A-MKL

$$\min_d G(d) = \frac{1}{2} \cdot \text{MMD}^2(d) + \theta \cdot J(d)$$

where

$$J(d) = \min_{w_m, b, \xi_i} \frac{1}{2} \left( \sum_{m=1}^M d_m \|w_m\|^2 + \lambda \|\beta\|^2 \right) + C \sum_{i=1}^n \xi_i$$

s.t. $y_i \cdot f^T(x) \geq 1 - \xi_i, \xi_i \geq 0$

- Dual form of $J(d)$

$$\min_{\alpha} -\alpha^T 1 + \frac{1}{2} (\alpha \circ y)^T \left( \sum_{m=1}^M d_m \tilde{K}_m \right) (\alpha \circ y)$$

s.t. $\alpha^T y = 0$, $0 \leq \alpha \leq C 1$

where

$$\tilde{K}_m(x_i, x_j) = K(x_i, x_j) + \frac{1}{\lambda} \sum_{p=1}^n f_p(x_i) f_p(x_j)$$

Solution:

Iteratively optimize $d$ and solve the SVM problem.

Image: Courtesy to D. Xu.

\textsuperscript{8}Duan et al. Visual Event Recognition in Videos by Learning from Web Data, CVPR’10.
Video event recognition

- **Goal:** Recognize consumer videos using *weakly labeled* web videos

![Diagram showing weakly labeled web videos, consumer videos, and a target classifier for consumer videos.](image)

- **Means and standard deviations of mean average precision (%) over six classes**

<table>
<thead>
<tr>
<th></th>
<th>SVM_T</th>
<th>SVM_AT</th>
<th>FR</th>
<th>A-SVM</th>
<th>MKL</th>
<th>DTSVM</th>
<th>A-MKL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP-(a)</td>
<td>42.32 ± 5.50</td>
<td>53.93 ± 5.58</td>
<td>49.98 ± 5.63</td>
<td>38.42 ± 7.93</td>
<td>47.19 ± 2.59</td>
<td>52.36 ± 1.88</td>
<td>57.14 ± 2.34</td>
</tr>
<tr>
<td>MAP-(b)</td>
<td>32.56 ± 2.08</td>
<td>24.73 ± 2.22</td>
<td>28.44 ± 2.61</td>
<td>24.95 ± 1.25</td>
<td>35.34 ± 1.55</td>
<td>31.07 ± 2.60</td>
<td>37.24 ± 1.58</td>
</tr>
<tr>
<td>MAP-(c)</td>
<td>42.00 ± 4.94</td>
<td>36.23 ± 3.37</td>
<td>44.11 ± 3.57</td>
<td>32.40 ± 4.99</td>
<td>46.92 ± 2.53</td>
<td>53.78 ± 2.99</td>
<td><strong>58.20 ± 1.87</strong></td>
</tr>
</tbody>
</table>

Image: Courtesy to D. Xu.

- **A-SVM** - Adaptive SVM, Yang *et al.* MM’07.
- **DTSVM** - Domain Transfer SVM, Duan, CVPR’09.
- **A-MKL** - Adaptive Multiple Kernel Learning, Duan *et al.* CVPR’10.
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Feature space transformation

Unsupervised feature transform
- Align pivot features (Biltzer et al. EMNLP’06, Pan et al. WWW’10)
- Manifold based methods (Pan et al. IJCAI’09, Gong et al. CVPR’12)
- Unsupervised subspace alignment (Fernando et al. ICCV’12)
- Stacked Marginalized Denoising Autoencoders (Chen et al. ICML’12)

Semi-supervised feature transform
- Metric learning based approaches (Zha et al. IJCAI’09, Saenko et al. ECCV’10, Hoffman et al. ECCV’12, Csurka et al. Task-CV’14)
- Semi-supervised Transfer Component Analysis (Pan et al. TNN’11)
Structural Correspondence Learning

- Identify pivot features by mutual information between features and domains.
- Build $P$ classifiers to predict the $P$ pivot features from remaining features.
- Project to the shared subspace (using the top $K$ eigenvectors).
- Concatenate with original features and train classifiers.

---

Blitzer et al., Domain Adaptation with Structural Correspondence Learning, EMNLP’06
Spectral Feature Alignment

- Bipartite graph to model correlations between pivot features and the others.
- Discover new shared features by spectral clustering on the graph.

---

Pan et al., Cross-Domain Sentiment Classification via Spectral Feature Alignment, WWW’10
Transfer Component Analysis\textsuperscript{11} (TCA)

\textbf{Diagram:}

\begin{itemize}
  \item Source
  \item Target
  \item Latent factors
  \item Temperature
  \item Signal properties
  \item Power of APs
  \item Building structure
  \item Cause the data distributions between domains different
\end{itemize}

\textbf{Equation:}

\begin{align*}
\min_W & \quad \text{tr}(W^T K L K W) + \text{tr}(W^T W) \\
\text{s.t.} & \quad W^T K H K W = I.
\end{align*}

\textbf{Result:}

\[ W^* \iff m \text{ leading eigenvectors of } (K L K + \lambda I)^{-1} K H K, \]
where \( m \leq n_S + n_T - 1. \)

\textbf{Image:} Courtesy to Pan.

\textsuperscript{11} Pan \textit{et al.}, Domain Adaptation via Transfer Component Analysis, IJCAI’09

G. Csurka, DA for Visual Applications
Geodesic Flow Sampling\textsuperscript{12} (GFS)

Apply PCA on source data ($S_1$ of rank $d$) and on the target ($S_2$ of rank $d$).

Geodesic path on the Grassman manifold $\mathbb{G}_{N,d}$ between $S_1$ and $S_2$.

Exponential flow $\psi(t') = Q \exp(t'B)J$ such that $Q^T S_1 = J$ and $J^T = [I_d \ 0_{N-d,d}]$.

Compute $B$ for intermediate subspaces varying $t \in [0, 1]$.

\textsuperscript{12}Gopalan \textit{et al.} Domain adaptation for object recognition: An unsupervised approach, ICCV’11.
Geodesic Flow Kernel\textsuperscript{13} (GFK)

\[
\text{Domain invariant features (infinite number of projections)}
\]

\[
z^{\infty} = [\Phi(0)^T \mathbf{x}, \ldots \Phi(t)^T \mathbf{x} \ldots \Phi(1)^T \mathbf{x}]
\]

\[
< z^\infty_i , z^\infty_j > = \int_0^1 (\Phi(t)^T \mathbf{x})^T (\Phi(t)^T \mathbf{x}) \, dt = \mathbf{x}_i^T \mathbf{G} \mathbf{x}_j
\]

\text{Image: Courtesy to Gong.}

\text{\textsuperscript{13}Gong et al., Geodesic flow kernel for unsupervised domain adaptation, CVPR’12.}

\textit{G. Csurka, DA for Visual Applications}
Subspace Alignment\textsuperscript{14} (SA)

\begin{itemize}
  \item $M^* = S_1'S_2$ corresponds to the “subspace alignment matrix”:
    $$M^* = \underset{M}{\text{argmin}} \| S_1 M - S_2 \|$$
  \item $X_a = S_1 S_1'S_2 = S_1 M^*$ projects the source data to the target subspace
  \item A natural similarity: $\text{Sim}(x_s, x_t) = x_s S_1 M^* S_1' x_t' = x_s A x_t'$
\end{itemize}

Image: Courtesy to Fernando.

\textsuperscript{14} Fernando et al. Unsupervised Visual Domain Adaptation Using Subspace Alignment, ICCV’13.
Marginalized Denoising Autoencoders\textsuperscript{15} (MDA)

- Learns a direct mapping that allow closed form solution for a given corruption.
- Generates many (ideally infinity) corruptions for each input.

\textsuperscript{15}Chen \textit{et al}., Marginalized Stacked Denoising Autoencoders for Domain Adaptation, ICML’12

G. Csurka, DA for Visual Applications
MDA¹⁶

Marginalizes out the corruption (convergence to expected values), hence \( W \) can be expressed in closed form as \( W^* = \mathbb{E}[P] \mathbb{E}[Q]^{-1} \), where:

\[
\mathbb{E}[P]_{ij} = S_{ij}q_j \quad \text{and} \quad \mathbb{E}[Q]_{ij} = \begin{cases} 
S_{ij}q_iq_j, & \text{if} \quad i \neq j \\
S_{ij}q_i, & \text{if} \quad i = j
\end{cases}
\]

with:

- \( q = [1 - p, \ldots, 1 - p, 1] \in \mathbb{R}^{n+1} \), \( p \) being the noise level and \( n \) the feature dimension
- \( S = XX^\top \) the covariance matrix of the uncorrupted data \( X \)
- \( X = [x_1, \ldots, x_m] \) where \( x_i = [x_i, 1] \) for the inputs \( x_i \), where the constant is never corrupted

¹⁶ Chen et al., Marginalized Stacked Denoising Autoencoders for Domain Adaptation, ICML’12

G. Csurka, DA for Visual Applications
Stacked MDA\textsuperscript{17} (SMDA)

Several MDA layers are stacked together

Applying nonlinearities between layers helps

- tangent-hyperbolic nonlinearities: \( h_t = \tanh(W^t h_{t-1}) \)

\textsuperscript{17}Chen \textit{et al.}, Marginalized Stacked Denoising Autoencoders for Domain Adaptation, ICML'12
Results on the OC10 (US sse) dataset

<table>
<thead>
<tr>
<th></th>
<th>C -&gt;A</th>
<th>D -&gt;A</th>
<th>W -&gt;A</th>
<th>A -&gt;C</th>
<th>D -&gt;C</th>
<th>W -&gt;C</th>
<th>A -&gt;D</th>
<th>C -&gt;D</th>
<th>W -&gt;D</th>
<th>A -&gt;W</th>
<th>C -&gt;W</th>
<th>D -&gt;W</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCA</td>
<td>46.7</td>
<td>39.6</td>
<td>40.2</td>
<td>40</td>
<td>34</td>
<td>33.7</td>
<td>39.1</td>
<td>41.4</td>
<td>77.5</td>
<td>40.1</td>
<td>36.2</td>
<td>80.4</td>
<td>45.74</td>
</tr>
<tr>
<td>GFS</td>
<td>36.8</td>
<td>32</td>
<td>27.5</td>
<td>35.3</td>
<td>29.4</td>
<td>21.7</td>
<td>30.7</td>
<td>32.6</td>
<td>54.3</td>
<td>31</td>
<td>30.6</td>
<td>66</td>
<td>35.66</td>
</tr>
<tr>
<td>GFK (PCA)</td>
<td>36.9</td>
<td>32.6</td>
<td>31.3</td>
<td>35.6</td>
<td>29.8</td>
<td>27.3</td>
<td>35.2</td>
<td>35.2</td>
<td>70.6</td>
<td>34.4</td>
<td>33.7</td>
<td>74.9</td>
<td>39.79</td>
</tr>
<tr>
<td>GFK (PLS)</td>
<td>40.4</td>
<td>36.1</td>
<td>35.5</td>
<td>37.9</td>
<td>32.7</td>
<td>29.3</td>
<td>35.1</td>
<td>41.1</td>
<td>71.2</td>
<td>35.7</td>
<td>35.8</td>
<td>79.1</td>
<td>42.49</td>
</tr>
<tr>
<td>SA (SVM)</td>
<td>46.1</td>
<td>42</td>
<td>39.3</td>
<td>39.9</td>
<td>35</td>
<td>31.8</td>
<td>38.8</td>
<td>39.4</td>
<td>77.9</td>
<td>39.6</td>
<td>38.9</td>
<td>82.3</td>
<td>45.92</td>
</tr>
<tr>
<td>SMDA</td>
<td>49.85</td>
<td>37.05</td>
<td>37.26</td>
<td>41.99</td>
<td>36.65</td>
<td>33.93</td>
<td>37.17</td>
<td>45.59</td>
<td>73.31</td>
<td>37.28</td>
<td>43.7</td>
<td>80.3</td>
<td>46.17</td>
</tr>
</tbody>
</table>

- **TCA** - Transfer Component Analysis, Pan et al. IJCAI’09.
- **GFS** - Geodesic Flow Sampling, Gopalan et al. ICCV’11.
- **GFK** - Geodesic Flow Kernel, B. Gong et al. CVPR’12.
- **SA** - Subspace Alignment, Fernando et al. ICCV’13.
- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen et al. ICML’12.
Results on AMT

- **SCL** - Structural Correspondence Learning, Blitzer et al., EMNLP’06.
- **SDA** - Deep learning approach - Glorot et al. ICML’11.
- **CODA** - Co-training for Domain Adaptation, Chen et al. NIPS’11.
- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen et al. ICML’12.

*Image: Courtesy to M. Chen.*
Information Theoretic Metric Learning\textsuperscript{18} (ITML)

(a) Domain shift problem  
(b) Pairwise constraints  
(c) Invariant space

Formulation (based on ITML [Davis et al., ICML'07])

\[
\min_{\mathbf{W} \succeq 0} \quad \text{Tr}(\mathbf{W}) - \log \det \mathbf{W} \\
\text{s.t.} \quad d^2_{\mathbf{W}}(x_i^s, x_j^t) \leq u, \forall (x_i^s, x_j^t) \in \text{SimilarSet} \\
\quad d^2_{\mathbf{W}}(x_i^s, x_j^t) \geq l, \forall (x_i^s, x_j^t) \in \text{DissimilarSet}
\]

\Rightarrow \text{Can be kernelized}

[18] Saenko et al., Adapting visual category models to new domains, ECCV 10

Image: Courtesy to Habrard.
The class label of a target is predicted based on

\[ p(c|x_i) = \frac{\sum_{d \in D} w_d e^{-\frac{1}{2} \| x_i - \mu_{d}^c \|}}{Z_i = \sum_{c'} \sum_{d} w_d e^{-\frac{1}{2} \| x_i - \mu_{d}^{c'} \|}} \]

where

- \( \mu_{d}^c \) is the mean of the class \( c \in C \) in the domain \( d \in D \)
- \( w_d \) are domain specific weights (we used \( w_{s_i} = 1 \) and \( w_t = 2 \))

\[ ^{19} \text{Csurka et al., Domain adaptation with a domain specific class means classifier. TASK-CV14} \]
Metric Learning for DSCMs\textsuperscript{20} (MLDSCM)

\begin{align*}
p(c|x_i) &= \frac{\sum_d w_d \mathcal{N}(W x_i, W \mu_d^c, \Sigma)}{\sum_{c'} \sum_d w_d \mathcal{N}(W x_i, W \mu_{d'}^{c'}, \Sigma)} \\
&= \frac{\sum_d w_d \exp \left( -\frac{1}{2} d_W(x_i, \mu_d^c) \right)}{\sum_{c'} \sum_d w_d \exp \left( -\frac{1}{2} d_W(x_i, \mu_{d'}^{c'}) \right)}
\end{align*}

with

\begin{itemize}
  \item domain-specific class means $\mu_d^c$,
  \item domain-specific weights $w_d$.
\end{itemize}

\textsuperscript{20}G. Csurka \textit{et al.} Domain Adaptation with a Domain Specific Class Means Classifier, Task-CV’14.
Semi-Supervised TCA\textsuperscript{21}

- $K$ is the (RBF) kernel matrix between the data points (both source and target).
- $L$ is the Laplacian of the affinity matrix $M_{i,j} = \exp(-d_{i,j}^2 / 2\sigma^2)$.
- $K_{yy}$ is a kernel matrix corresponding to source labels.
- $L$ integrates the normalizations $1/N_S^2$, $1/N_T^2$ and $1/N_SN_T$ and $H$ is a centering matrix.

\textbf{Algorithm:} generalized eigen decomposition

Image: Courtesy to Pan.

\textsuperscript{21}S.J. Pan \textit{et al.}, Domain Adaptation via Transfer Component Analysis, TNN’11.
Results on the OC10 (SS sse) dataset

- **ITML** - Information Theoretic Metric Learning, Saenko et al., ECCV’10
- **SSTCA** - Semi-Supervised TCA, Pan et al., TNN’11.

<table>
<thead>
<tr>
<th></th>
<th>C -&gt; A</th>
<th>D -&gt; A</th>
<th>W -&gt; A</th>
<th>A -&gt; C</th>
<th>D -&gt; C</th>
<th>W -&gt; C</th>
<th>A -&gt; D</th>
<th>C -&gt; D</th>
<th>W -&gt; D</th>
<th>A -&gt; W</th>
<th>C -&gt; W</th>
<th>D -&gt; W</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITML</td>
<td>33.7</td>
<td>30.3</td>
<td>32.3</td>
<td>27.3</td>
<td>22.5</td>
<td>21.7</td>
<td>33.7</td>
<td>35</td>
<td>51.3</td>
<td>36</td>
<td>34.7</td>
<td>55.6</td>
<td>34.51</td>
</tr>
<tr>
<td>MLDSCM</td>
<td>50.64</td>
<td>48.76</td>
<td>48.43</td>
<td>34.89</td>
<td>34.24</td>
<td>33.42</td>
<td>62.05</td>
<td>61.57</td>
<td>64.65</td>
<td>66.08</td>
<td>65.06</td>
<td>71.47</td>
<td>53.44</td>
</tr>
<tr>
<td>SSTCA</td>
<td>47.1</td>
<td>40.1</td>
<td>41.5</td>
<td>40.4</td>
<td>34.2</td>
<td><strong>33.5</strong></td>
<td>39</td>
<td>41.7</td>
<td>77.8</td>
<td>41.1</td>
<td>36.2</td>
<td><strong>80.5</strong></td>
<td>46.09</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Methods combining several of the above ideas

- Joint feature and parameter adaptation
  - Max-Margin Domain Transforms (MMDT) of Hoffman et al. ICLR13
  - Naive Bayes NN based DA (NBNN-DA) of Tommasi and Caputo, ICCV13
  - Domain Invariant Projection (DIP-CC) of M. Baktashmotlagh et al. ICCV13
  - Joint Distribution Adaptation (JDA) of Long et al. ICCV 14
  - Transfer Joint Matching (TJM) of Long et al. ICCV 14
  - Optimal Transport for Domain Adaptation (OTDA) of Courty et al. PAMI 15

- Joint feature or instance selection and parameter adaptation
  - Feature selection and subspace learning (FSSL) of Gu et al. IJCAI 11
  - Landmark Selection (LM) of Gong et al. ICML13
  - Landmarks Selection-based SA (LSSA) of Aljundi CVPR15

- Joint instance selection, feature transform and parameter adaptation
  - Adaptive Transductive Transfer Machines (ATTM) of Farajidavar et al. BMVC 14
  - Statistically Invariant Embedding (SIE-CC) of M. Baktashmotlagh et al. CVPR 14
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   - Instance reweighing methods
   - Parameter based methods
   - Feature transformation-based methods

3. Combined methods
   - Joint feature transform and parameter adaptation
   - Joint feature/instance selection and feature transform
   - Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Max-Margin Domain Transforms\textsuperscript{22} (MMDT)

\[\min_{w,\theta} \frac{1}{2}\|w\|^2_F + \frac{1}{2}\|\Theta\|^2_F + \lambda \mathcal{L}(w, \Theta, z, h) + \lambda_x \mathcal{L}(\Theta, X, y)\]

\text{regularizer} \quad \text{loss}

Image: Courtesy to Hoffman.

\textsuperscript{22}Hoffman et al., Max-margin transforms for visual domain adaptation, ICLR'13.
Naive Bayes NN based DA \(^{23}\) (NBNN-DA)

Iteratively combine metric learning and NBNN-based sample selection to:

- adjust the image-to-class distances by tuning the per class metrics
- iteratively making the metric progressively more suitable for the target

\(23\) Tommasi and Caputo, Frustratingly Easy NBNN Domain Adaptation, ICCV’13.
Transfer Joint Matching\textsuperscript{24} (TJM)

\textbf{Algorithm 1: TJM: Transfer Joint Matching}

\begin{enumerate}
\item \textbf{Input:} Data $X$; \#subspace bases $k$, regularization parameter $\lambda$.
\item \textbf{Output:} Adaptation matrix $A$, embedding $Z$, adaptive classifier $f$.
\item \textbf{begin}
\item \hspace{1em} Compute MMD matrix $M$ by Equation (5), and kernel matrix $K$ by $K_{ij} \leftarrow K(x_i, x_j)$ where $K(\cdot, \cdot)$ is a predefined kernel.
\item \hspace{1em} Set $M \leftarrow M/\|M\|_F$, $G \leftarrow I$.
\item \hspace{1em} \textbf{repeat}
\item \hspace{2em} Solve the generalized eigendecomposition problem in Equation (9) and select the $k$ smallest eigenvectors to construct the adaptation matrix $A$, and $Z \leftarrow A^T K$.
\item \hspace{2em} Update the sub-gradient matrix $G$ by Equation (10).
\item \hspace{1em} \textbf{until} Convergence
\item \hspace{1em} Return an adaptive classifier $f$ trained on $\{A^T k_i, y_i\}_{i=1}^{n_s}$.
\end{enumerate}

\textsuperscript{24}Long \textit{et al.} Transfer Joint Matching for Unsupervised Domain Adaptation, CVPR’14.

G. Csurka, DA for Visual Applications
Joint Distribution Adaptation\textsuperscript{25} (JDA)

\begin{algorithm}
\textbf{Algorithm 1: JDA: Joint Distribution Adaptation}
\begin{algorithmic}
\State \textbf{Input:} Data $X$, $y_s$; \#subspace bases $k$, regularization parameter $\lambda$.
\State \textbf{Output:} Adaptation matrix $A$, embedding $Z$, adaptive classifier $f$.
\State \textbf{begin}
\State \quad Construct MMD matrix $M_0$ by Eq. (4), set $\{M_c := 0\}_{c=1}^C$.
\State \quad \textbf{repeat}
\State \quad \quad Solve the generalized eigendecomposition problem in Equation (10) and select the $k$ smallest eigenvectors to construct the adaptation matrix $A$, and $Z := A^T X$.
\State \quad \quad Train a standard classifier $f$ on $\{(A^T x_i, y_i)\}_{i=1}^{n_s}$ to update pseudo target labels $\{\hat{y}_j := f(A^T x_j)\}_{j=n_s+1}^{n_s+n_t}$.
\State \quad \quad Construct MMD matrices $\{M_c\}_{c=1}^C$ by Equation (6).
\State \quad \textbf{until} Convergence
\State \quad Return an adaptive classifier $f$ trained on $\{A x_i, y_i\}_{i=1}^{n_s}$.
\State \textbf{end}
\end{algorithmic}
\end{algorithm}

\textsuperscript{25}M. Long \textit{et al.} Transfer Feature Learning with Joint Distribution Adaptation, ICCV’14
Optimal Transport for Domain Adaptation\textsuperscript{26} (OTDA)

\textbf{Algorithm 1} Conditional gradient splitting (CGS)

1. Initialize $k = 0$ and $\gamma^0 \in \mathcal{P}$
2. repeat
3. With $G \in \nabla f(\gamma^k)$, solve
   \[ \gamma^* = \arg\min_{\gamma \in \mathcal{B}} \langle \gamma, G \rangle_{\mathcal{F}} + g(\gamma) \]
4. Find the optimal step with $\Delta \gamma = \gamma^* - \gamma^k$
   \[ \alpha^k = \arg\min_{0 \leq \alpha \leq 1} f(\gamma^k + \alpha \Delta \gamma) + g(\gamma^k + \alpha \Delta \gamma) \]
5. $\gamma^{k+1} \leftarrow \gamma^k + \alpha^k \Delta \gamma$, set $k \leftarrow k + 1$
6. until Convergence

\text{Image: Courtesy to Courty.}

\textsuperscript{26}N. Courty \textit{et al.}, Optimal Transport for Domain Adaptation, CoRR’15.
Domain Invariant Projection\(^\text{(DIP-CC)}\)

- Optimizing the MMD on the Grassman manifold \(G(d, D)\)

\[
D_{\mathcal{H}}(W^\top X_S, W^\top X_T) = \left\| \frac{1}{N_s} \sum_{i=1}^{N_s} \phi(W^\top x^s_i) - \frac{1}{N_t} \sum_{j=1}^{N_t} \phi(W^\top x^t_j) \right\|_{\mathcal{H}}
\]

where \(W\) is a point on \(G\) with the constraint \(W^\top W = I\).

- Adding term to encourage Class Clustering (CC):

\[
\sum_{c=1}^{C} \sum_{i=1}^{n_c} \|W^\top (x^s_{i,c} - \mu_c)\|^2
\]

- The whole yielding to the optimization problem:

\[
W^* = \arg\min_W \text{Tr}(K_W L) + \lambda \sum_{c=1}^{C} \sum_{i=1}^{n_c} \|W^T(x^s_{i,c} - \mu_c)\|^2 \\
\text{s.t. } W^\top W = I,
\]

\[
K_W = \begin{bmatrix} K_{s,s} & K_{s,t} \\ K_{t,s} & K_{t,t} \end{bmatrix} \in \mathbb{R}^{(n+m) \times (n+m)}
\]

\[
L_{ij} = \begin{cases} 1/n^2 & i, j \in S \\ 1/m^2 & i, j \in T \\ -1/(nm) & \text{otherwise} \end{cases}
\]

\(27\) M. Baktashmotlagh et al., Unsupervised Domain Adaptation by Domain Invariant Projection, ICCV’13.
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   **Joint feature/instance selection and feature transform**
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Feature selection and subspace learning²⁸ (FSSL)

Algorithm 2 Joint Feature Selection and Subspace Learning (Situation 2)

**Initialize:** $G_0 = I$, $t = 0$ and $\mu$;
Compute $Y$ based on $WY = \Lambda DY$;
repeat
    Compute $A_{t+1} = G_t^{-1}X(X^TG_t^{-1}X + \frac{1}{2\mu}I)^{-1}Y$;
    Compute $G_{t+1}$ based on $A_{t+1}$;
    $t = t + 1$;
until convergence

Landmark Selection\(^{29}\)

*Landmarks* are labeled source instances distributed similarly to the target domain.

Identifying landmarks:

\[
P_L(\text{landmarks}) \approx P_T(\text{target})
\]

\[
\min_{\text{landmarks}} d(P_L, P_T)
\]

[Gong et al., ICML’13]

Convex relaxation

\[
\min_{\{\alpha_m\}} \left\| \frac{1}{\sum_i \alpha_i} \sum_{m=1}^M \alpha_m \phi(x_m) - \frac{1}{N} \sum_{n=1}^N \phi(x_n) \right\|^2_H
\]

\[
\beta_m = \frac{\alpha_m}{\sum_i \alpha_i}
\]

\[
\min_\beta \beta^T K^s \beta - \frac{2}{N} \beta^T K^{st} 1
\]

Image: Courtesy to Gong.

\(^{29}\) Gong *et al.*, Connecting the dots with landmarks: Discriminatively learning domain invariant features for unsupervised domain adaptation, ICML’13.
Landmarks Selection-based SA\textsuperscript{30} (LSSA)

- Landmark selection using a Gaussian Kernels and overlap between the probability densities:

\[
\text{overlap}(\mu_S, \sigma_S; \mu_T, \sigma_T) = \frac{\mathcal{N}(\mu_S, -\mu_T|0, (\sigma_S + \sigma_T)^2)}{\mathcal{N}(0|0, (\sigma_S + \sigma_T))}
\]

- Subspace Alignment using the selected landmarks and linear SVM classifiers.

---

\textsuperscript{30} Aljundi \textit{et al.}, Landmarks-based Kernelized Subspace Alignment for Unsupervised Domain Adaptation, CVPR‘15.
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Transductive Transfer Machines\textsuperscript{31} (ATTM)

\textbf{TTM Diagram}

\textbf{ATTM Algorithm}

\begin{itemize}
  \item \textbf{Input:} $X^{src}$, $Y^{src}$, $X^{trg}$
  \item \textbf{Output:} $Y^{trg}$
  \item 1. Search for the shared subspace between the two domains
  \item 2. Adjust the marginal distribution mismatch between the two domains
  \item 3. Select the appropriate classifier, if it is kernel-based, tune $\sigma$ using
  \item \textbf{while} $T < 10$ and $|D^{global}(G'(X^{src}), X^{trg})| > \text{threshold}$ \textbf{do}
  \item 4. Find the feature-wise TST transformation
  \item 5. Transform the source domain clusters
  \item 6. Retrain the classifier using the transformed source
  \item \textbf{end while}
\end{itemize}

Image: Courtesy to deCampos.

\textsuperscript{31}Farajidavar \textit{et al}., Adaptive Transductve Transfer Machines, BMVC’14.
Statistically Invariant Embedding (SIE-CC)

- Statistically Invariant Sample Selection

\[
\min_{\beta} \sum_{i=1}^{n_s} \beta_i \left( \sqrt{T(x_i^s)} - \sqrt{1 - T(x_i^s)} \right)^2 + \frac{1}{n_t} \sum_{i=1}^{n_t} \left( \sqrt{T(x_i^t)} - \sqrt{1 - T(x_i^t)} \right)^2
\]

s.t. \( \beta_i \in [0, 1] \); \( \sum_{i=1}^{n_s} \beta_i = 1 \); \( \sum_{i=1}^{n_s} \beta_i y_i,c = \frac{1}{n_s} \sum_{i=1}^{n_s} y_i,c \forall c \)

- Statistically Invariant Embedding

\[
\min_{W} D_H(W^\top X_S, W^\top X_T) + \lambda \sum_{c=1}^{C} \sum_{i=1}^{n_c} \|W^\top (x_{i,c}^s - \mu_c)\|^2
\]

s.t. \( W^\top W = I \)

\[32\] Baktashmotlagh et al., Domain Adaptation on the Statistical Manifold, CVPR’14.
Digit recognition results

- **GFK** - Geodesic Flow Kernel, B. Gong *et al.* CVPR’12.
- **TCA** - Transfer Component Analysis, Pan *et al.* IJCAI’09.
- **FFSL** - Feature selection and subspace learning, Gu *et al.* IJCAI’11.
- **OTDA** - Optimal Transport for Domain Adaptation, Courty *et al.* CoRR’15.

<table>
<thead>
<tr>
<th></th>
<th>GFK</th>
<th>TCA</th>
<th>FSSL</th>
<th>TJM</th>
<th>JDA</th>
<th>OTDA</th>
<th>ATTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>U -&gt; M</td>
<td>46.45</td>
<td>51.05</td>
<td>51.45</td>
<td>52.25</td>
<td>59.65</td>
<td>58.3</td>
<td><strong>61.15</strong></td>
</tr>
<tr>
<td>M -&gt; U</td>
<td>67.22</td>
<td>56.28</td>
<td>57.44</td>
<td>63.28</td>
<td>67.28</td>
<td>69.39</td>
<td><strong>77.9</strong></td>
</tr>
</tbody>
</table>

G. Csurka, DA for Visual Applications
Object recognition (OC10 US ase)

<table>
<thead>
<tr>
<th>Method</th>
<th>C -&gt;A</th>
<th>W -&gt;A</th>
<th>A -&gt;C</th>
<th>W -&gt;C</th>
<th>A -&gt;D</th>
<th>C -&gt;D</th>
<th>W -&gt;D</th>
<th>A -&gt;W</th>
<th>C -&gt;W</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMDA</td>
<td>52.8</td>
<td>35.3</td>
<td>41.9</td>
<td>32.3</td>
<td>37.8</td>
<td>47.2</td>
<td>77.9</td>
<td>37</td>
<td>49.8</td>
<td>45.79</td>
</tr>
<tr>
<td>SA</td>
<td>52.7</td>
<td>39.4</td>
<td>41.6</td>
<td>34.7</td>
<td>46.4</td>
<td>49</td>
<td>78.9</td>
<td>40.7</td>
<td>42.7</td>
<td>47.34</td>
</tr>
<tr>
<td>LSSA</td>
<td>58.4</td>
<td>39.4</td>
<td>44.8</td>
<td>34.7</td>
<td>42.4</td>
<td>54.1</td>
<td>87.2</td>
<td>42.4</td>
<td>48.1</td>
<td>50.17</td>
</tr>
<tr>
<td>LS</td>
<td>56.7</td>
<td>40.2</td>
<td>45.5</td>
<td>35.4</td>
<td>47.1</td>
<td>57.3</td>
<td>75.2</td>
<td>46.1</td>
<td>49.5</td>
<td>50.3</td>
</tr>
<tr>
<td>TJM</td>
<td>58.6</td>
<td>40.8</td>
<td>45.7</td>
<td>34.8</td>
<td>42</td>
<td>49</td>
<td>83.4</td>
<td>42</td>
<td>48.8</td>
<td>49.45</td>
</tr>
<tr>
<td>JDA</td>
<td>44.8</td>
<td>32.8</td>
<td>39.4</td>
<td>31.2</td>
<td>39.5</td>
<td>45.2</td>
<td>89.2</td>
<td>38</td>
<td>41.7</td>
<td>44.62</td>
</tr>
<tr>
<td>OTDA</td>
<td>48</td>
<td>40</td>
<td>38.6</td>
<td>37.2</td>
<td>45</td>
<td>45.2</td>
<td>93</td>
<td>41.4</td>
<td>44.7</td>
<td>48.08</td>
</tr>
<tr>
<td>ATTM</td>
<td>60.8</td>
<td>39.7</td>
<td>42.9</td>
<td>34</td>
<td>31.8</td>
<td>50.3</td>
<td>89.2</td>
<td>50.5</td>
<td>38</td>
<td>48.59</td>
</tr>
<tr>
<td>DIP-CC</td>
<td>58.7</td>
<td>40.9</td>
<td>47.2</td>
<td>37.2</td>
<td>49</td>
<td>61.2</td>
<td>91.7</td>
<td>47.8</td>
<td>58</td>
<td>54.63</td>
</tr>
<tr>
<td>SIE-CC</td>
<td>57.6</td>
<td>42.4</td>
<td>47.6</td>
<td>36.2</td>
<td>49</td>
<td>61.2</td>
<td>93</td>
<td>47.8</td>
<td>57.3</td>
<td>54.68</td>
</tr>
</tbody>
</table>

- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen et al. ICML’12.
- **SA** - Subspace Alignment, Fernando et al. ICCV’13.
- **LSSA** - Landmarks Selection-based SA, Aljundi et al. CVPR’15.
- **LS** - Landmark Selection, Gong et al. ICML’13.
- **TJM** - Transfer Joint Matching, Long et al. CVPR’14.
- **JDA** - Joint Distribution Adaptation, long et al. ICCV’14.
- **OTDA** - Optimal Transport for Domain Adaptation, Courty et al. CoRR’15.
- **ATTM** - Adaptive Transductive Transfer Machines, Farajidavar et al., BMVC’14.
- **DIP-CC** Domain Invariant Projection, Baktashmotlagh et al. ICCV’13.
- **SIE-CC** - Statistically Invariant Embedding, Baktashmotlagh et al. CVPR’14.
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Example: cross-lingual text categorization

- Experiments on the Reuters multilingual dataset

<table>
<thead>
<tr>
<th>Source Domain</th>
<th>SVM_T</th>
<th>KCCA</th>
<th>HeMap</th>
<th>DAMA</th>
<th>ARC-t</th>
<th>HFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>72.6 ± 2.3</td>
<td>71.4 ± 3.2</td>
<td>65.7 ± 3.1</td>
<td>72.4 ± 2.4</td>
<td>72.9 ± 2.0</td>
<td>75.3 ± 1.7</td>
</tr>
<tr>
<td>French</td>
<td>72.8 ± 2.8</td>
<td>72.8 ± 2.8</td>
<td>64.2 ± 4.2</td>
<td>72.8 ± 2.0</td>
<td>73.5 ± 1.8</td>
<td>75.7 ± 1.6</td>
</tr>
<tr>
<td>German</td>
<td>73.8 ± 2.2</td>
<td>73.8 ± 2.2</td>
<td>64.6 ± 3.6</td>
<td>72.9 ± 2.3</td>
<td>74.7 ± 1.6</td>
<td>76.1 ± 1.5</td>
</tr>
<tr>
<td>Italian</td>
<td>73.8 ± 2.1</td>
<td>65.8 ± 2.3</td>
<td>73.3 ± 2.1</td>
<td>74.0 ± 2.0</td>
<td>75.8 ± 1.8</td>
<td></td>
</tr>
</tbody>
</table>

- Means and standard deviations of classification accuracies (%) of all methods on the Reuters multilingual dataset by using 10 labeled training samples per class from the target domain Spanish. Results in boldface are significantly better than the others, judged by the t-test with a significance level at 0.05.

Image: Courtesy to Dong Xu.
Heterogeneous Feature Augmentation\textsuperscript{33} (HFA)

- **Objective**
  - The dual form is similar to SVM with a different kernel.

\[
\begin{align*}
\min \max_{P, Q} & \quad 1'_{n_s+n_t} \alpha - \frac{1}{2} (\alpha \circ y)'K_{P, Q}(\alpha \circ y) \\
\text{s.t.} & \quad y'\alpha = 0, \quad 0_{n_s+n_t} \leq \alpha \leq C1_{n_s+n_t}, \\
& \quad \|P\|_F^2 \leq \lambda_P, \quad \|Q\|_F^2 \leq \lambda_Q.
\end{align*}
\]

- Global optimum can be solved using the method similarly as in MKL (see our T-PAMI 2014 work for more details).

\[
\begin{array}{c}
\phi_s(x^s) = \begin{bmatrix} Px^s \\ x^s \\ 0_{d_t} \end{bmatrix} \\
\phi_t(x^t) = \begin{bmatrix} Qx^t \\ 0_{d_s} \\ x^t \end{bmatrix}
\end{array}
\]

Image: Courtesy to Dong Xu.

- Generalizes the feature replication method of Daume III, ACL 07.

\textsuperscript{33}Duan et al. Learning with Augmented Features for Heterogeneous Domain Adaptation, CVPR12.
Dictionary-based Approaches

- Each domain has its own projection matrix.
- Can be used when features in source and target are different.
- Generalizes well to multiple sources.

Image: Courtesy to S. Shekhar.

G. Csurka, DA for Visual Applications
Shared Domain-adapted Dictionary Learning (SDDL)

- Main idea:

\[
\{D^*, \tilde{W}^*, \tilde{X}^*\} = \arg\min_{D,\tilde{W},\tilde{X}} c_1(D, \tilde{W}, \tilde{X}) + \lambda c_2(\tilde{W}) \\
\text{s.t. } W_i W_i^T = I, \ i = 1, 2 \text{ and } ||\tilde{x}_j||_0 \leq T_0, \forall j
\]

\[
\tilde{W} = [W_1 \ W_2], \ \tilde{Y} = \begin{pmatrix} S & 0 \\ 0 & T_l \end{pmatrix}, \text{ and } \tilde{X} = [X_1 \ X_2].
\]

\[
c_1(D, \tilde{W}, \tilde{X}) = ||\tilde{W}\tilde{Y} - D\tilde{X}||_F^2,
\]

\[
c_2(\tilde{W}) = -\text{trace}((\tilde{W}\tilde{Y})(\tilde{W}\tilde{Y})^T)
\]

- Its kernelized extension

\[
c_1(\tilde{A}, \tilde{B}, \tilde{X}) = ||\tilde{A}^T\mathcal{K}(I - \tilde{B}\tilde{X})||_F^2 + \\
\mu||\tilde{A}^T\mathcal{K}(I - \tilde{B}\tilde{X}_{\text{in}})||_F^2 + \nu||\tilde{A}^T\mathcal{K}\tilde{B}\tilde{X}_{\text{out}}||_F^2,
\]

\[
c_2(\tilde{A}) = -\text{trace}((\tilde{A}^T\mathcal{K})(\tilde{A}^T\mathcal{K})^T)
\]

---

34 Shekhar et al., Generalized Domain-Adaptive Dictionaries, CVPR 13.
Results on the OC10 (SS sse)

- **FDDL** Fisher Discrimination Dictionary Learning, Yang et al. ICCV’11.
- **SDDL** Shared Domain-adapted Dictionary Learning, Shekhar et al. CVPR 13.
- **DIP-CC** Domain Invariant Projection, Baktashmotlagh et al. ICCV’13.

<table>
<thead>
<tr>
<th></th>
<th>C -&gt;A</th>
<th>D -&gt;A</th>
<th>W -&gt;A</th>
<th>A -&gt;C</th>
<th>W -&gt;C</th>
<th>C -&gt;D</th>
<th>A -&gt;W</th>
<th>D -&gt;W</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>FDDL</td>
<td>39.3</td>
<td>36.7</td>
<td>41.1</td>
<td>24.3</td>
<td>22.9</td>
<td>55</td>
<td>50.4</td>
<td>65.9</td>
<td>38.5</td>
</tr>
<tr>
<td>SDDL</td>
<td>49.5</td>
<td>48.9</td>
<td>49.4</td>
<td>27.4</td>
<td>29.7</td>
<td>76.7</td>
<td>72</td>
<td>72.6</td>
<td>53.3</td>
</tr>
<tr>
<td>MLDSCM</td>
<td>50.64</td>
<td>48.76</td>
<td>48.43</td>
<td>34.89</td>
<td>33.42</td>
<td>61.57</td>
<td>66.08</td>
<td>71.47</td>
<td>51.9</td>
</tr>
<tr>
<td>DIP-CC</td>
<td>61.8</td>
<td>56.9</td>
<td>53.4</td>
<td>47.8</td>
<td>43.6</td>
<td>65.8</td>
<td>72.5</td>
<td>89.1</td>
<td>61.36</td>
</tr>
</tbody>
</table>
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Multi-sources results on the OFF31 (SS sse)


<table>
<thead>
<tr>
<th></th>
<th>A,D-&gt;W</th>
<th>A,W-&gt;D</th>
<th>D,W-&gt;A</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSCM</td>
<td>58.21</td>
<td>57.65</td>
<td>21.67</td>
<td>45.84</td>
</tr>
<tr>
<td>SDDL</td>
<td>57.8</td>
<td>56.7</td>
<td>24.1</td>
<td>46.2</td>
</tr>
<tr>
<td>BGFS</td>
<td>64.5</td>
<td>51.3</td>
<td>38.4</td>
<td>51.4</td>
</tr>
<tr>
<td>HL2L</td>
<td>66.1</td>
<td>67.9</td>
<td>25.8</td>
<td>53.27</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A,D-&gt;W</th>
<th>A,W-&gt;D</th>
<th>D,W-&gt;A</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSCM</td>
<td>45.04</td>
<td>35.51</td>
<td>12.12</td>
<td>30.89</td>
</tr>
<tr>
<td>GFS</td>
<td>37.5</td>
<td>33.9</td>
<td>31.5</td>
<td>34.3</td>
</tr>
<tr>
<td>SSF</td>
<td>47.8</td>
<td>49.6</td>
<td>40.2</td>
<td>45.87</td>
</tr>
</tbody>
</table>

![Images](https://example.com/images.png)

G. Csurka, DA for Visual Applications
Boosted Geodesic Flow Sampling\textsuperscript{35} (BGFS)

- Intermediate "domains" between source and target on the Grassman Manifold
- Multi-domain adaptation using Karcher mean of domains
- Jointly learn domain shift features and classifiers with AdaBoost

\textsuperscript{35} Gopalan et al. Unsupervised Adaptation Across Domain Shifts By Generating Intermediate Data Representations, CoRR'15.
Spline Flow Sampling \(^{36}\) (SSF)

\[\alpha(t) : [0, T] \mapsto M\] the rolling geodesic curve on the manifold \(M\)

\[\beta(t) : [0, T] \mapsto \mathcal{V}\] the spline curve on the tangent space \(\mathcal{V}\)

\[\gamma(t) : [0, T] \mapsto M\] the spline curve on the manifold \(M\)

---

\(^{36}\) Caseiro et al. Beyond the shortest path: Unsupervised Domain Adaptation by Sampling Subspaces along the Spline Flow, CVPR 15.
High-level Learning to Learn (H-L2L)

Boosting approach to learn $\beta^c_y$ where the weak learners are:

$$s(x, y) = \beta^0 w^0_0 \phi^0_0(x, y) + \sum_{c=1}^{N_c} \beta^c_y w^c_y \phi^c_y(s_S(x, c), y)$$

- $s_S(x, c)$ the score of $x$ with the source classifier $c$
- $\phi^c_y \mapsto Y \times Y$ is the $y^{th}$ score mapping corresponding to the $c^{th}$ source model
- $w^c_y$ the $y^{th}$ source model in predicting that $x$ belongs to class $c$
- $\phi^0_0(x, y)$ is the feature mapping for the original input features

---

37 Patricia and Caputo, Learning to Learn, from Transfer Learning to Domain Adaptation: A Unifying Perspective, CVPR 14.
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
What about deep features?

High-non linearity makes these features more invariant across domains!
Deep convolutional activation features\(^\text{38}\) (DeCAF)

- CNN model pre-trained on big data (ImageNet)
- use convolutional activation layers as features.

---

\(^\text{38}\) Donuahe et al. DeCAF: A deep convolutional activation feature for generic visual recognition, ICML 14

---

G. Csurka, DA for Visual Applications
Compared to bag of visual-words\textsuperscript{39} (BOV)

DeCAF provides better category level clustering than SURF BOV


Image: Courtesy to Donuahe.
Results on OFF31 (US sse)

<table>
<thead>
<tr>
<th>(D)</th>
<th>bike</th>
<th>desk chair</th>
</tr>
</thead>
<tbody>
<tr>
<td>dslr</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(A)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>amazon</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(W)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>webcam</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A-&gt;W</th>
<th>D-&gt;W</th>
<th>W-&gt;D</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>42.8</td>
<td>86.4</td>
<td>88.6</td>
<td>72.6</td>
</tr>
<tr>
<td>SVM</td>
<td>63.05</td>
<td>95.04</td>
<td>96.74</td>
<td>84.94</td>
</tr>
<tr>
<td>NCM</td>
<td>64.76</td>
<td>92.17</td>
<td>94.37</td>
<td>83.77</td>
</tr>
<tr>
<td>SMDA</td>
<td>68.56</td>
<td>92.92</td>
<td>95.01</td>
<td>85.5</td>
</tr>
<tr>
<td>GFK</td>
<td>46.8</td>
<td>87.2</td>
<td>88.1</td>
<td>74.03</td>
</tr>
<tr>
<td>TCA</td>
<td>44.6</td>
<td>89</td>
<td>87.9</td>
<td>73.83</td>
</tr>
<tr>
<td>SA</td>
<td>47.2</td>
<td>91.8</td>
<td>92.4</td>
<td>77.13</td>
</tr>
</tbody>
</table>

- **NN** - Nearest Neighbor classifier (no adaptation).
- **SVM** - Linear SVM classifier (no adaptation).
- **NCM** - Nearest Class Means classifier (DSCM with single domain, no adaptation).
- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen et al. ICML’12.
- **GFK** - Geodesic Flow Kernel, B. Gong et al. CVPR’12.
- **TCA** - Transfer Component Analysis, Pan et al. IJCAI’09.
- **SA** - Subspace Alignment, Fernando et al. ICCV’13.
Results on OC10 (US sse)

- **NN** - Nearest Neighbor classifier (no adaptation).
- **SVM** - Linear SVM classifier (no adaptation).
- **NCM** - Nearest Class Means classifier (DSCM with single domain, no adaptation).
- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen et al. ICML’12.
- **JDA** - Joint Distribution Adaptation, long et al. ICCV’14.
- **ATTM** - Adaptive Transductive Transfer Machines, Farajidavar et al., BMVC’14.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NN</strong></td>
<td>85.7</td>
<td>66.1</td>
<td>74.52</td>
<td>70.35</td>
<td>64.97</td>
<td>57.29</td>
<td>60.37</td>
<td>62.53</td>
<td>98.73</td>
<td>52.09</td>
<td>62.73</td>
<td>89.15</td>
<td>70.33</td>
</tr>
<tr>
<td><strong>SVM</strong></td>
<td><strong>94.61</strong></td>
<td>85.28</td>
<td>87.4</td>
<td>88.47</td>
<td>86.79</td>
<td>88.98</td>
<td>86</td>
<td>87.28</td>
<td>100</td>
<td>84.81</td>
<td>87.28</td>
<td>98.4</td>
<td>89.61</td>
</tr>
<tr>
<td><strong>NCM</strong></td>
<td>94.07</td>
<td>87.55</td>
<td>87.4</td>
<td>88.2</td>
<td>87.17</td>
<td>85.04</td>
<td>87.01</td>
<td>90.3</td>
<td>99.21</td>
<td>85.82</td>
<td>90.52</td>
<td>98.49</td>
<td>90.6</td>
</tr>
<tr>
<td><strong>SMDA</strong></td>
<td>93.86</td>
<td><strong>90.19</strong></td>
<td>88.98</td>
<td><strong>89.48</strong></td>
<td><strong>87.92</strong></td>
<td>88.98</td>
<td><strong>89.48</strong></td>
<td><strong>93.64</strong></td>
<td>99.21</td>
<td><strong>89.02</strong></td>
<td><strong>93.32</strong></td>
<td><strong>99.25</strong></td>
<td><strong>91.94</strong></td>
</tr>
<tr>
<td><strong>JDA</strong></td>
<td>89.77</td>
<td>83.73</td>
<td>86.62</td>
<td>82.28</td>
<td>78.64</td>
<td>80.25</td>
<td>83.53</td>
<td><strong>100</strong></td>
<td>90.19</td>
<td>85.13</td>
<td>91.44</td>
<td>98.98</td>
<td>87.55</td>
</tr>
<tr>
<td><strong>ATTM</strong></td>
<td>92.17</td>
<td>90.84</td>
<td><strong>92.99</strong></td>
<td>86.55</td>
<td>89.15</td>
<td><strong>90.45</strong></td>
<td>83.44</td>
<td>92.27</td>
<td><strong>100</strong></td>
<td>82.28</td>
<td>91.65</td>
<td>98.98</td>
<td>90.9</td>
</tr>
</tbody>
</table>

G. Csurka, DA for Visual Applications
Interpolating between Domains\textsuperscript{40} (DLID)

- generate intermediate datasets $D_p$ by mixing target and source
- unsupervised deep nonlinear extractor $F_{W_p}$ learned on
- train classifiers on the concatenated features $Z_p^i = F_{W_i}(X^i)$.

\textsuperscript{40} S. Chopra \textit{et al}. Deep Learning for domain adaptation by Interpolating between Domains, RL-WS ICML13.

G. Csurka, DA for Visual Applications
Deep Domain Confusion \(^{41}\) (DDC)

Minimizing the loss

\[
\mathcal{L} = \mathcal{L}_C(X_L, y) + \lambda \text{MMD}^2(X_S, X_T)
\]

Deep Adaptation Networks\textsuperscript{42} (DAN)

- first 3 convolutional layers are kept frozen
- next 2 convolutional layers refined with the current dataset
- deeply adapt fc6-fc8 using Multiple Kernel MMD

\textsuperscript{42} M. Long \textit{et al.}, Learning Transferable Features with Deep Adaptation Networks, CoRR’15.
Domain Adaptation by Backpropagation\textsuperscript{43} (DAB)

\begin{itemize}
  \item $L_y$ is the loss for label prediction (e.g. multinomial),
  \item $L_d$ is the loss for the domain classification (e.g. logistic),
\end{itemize}

\textsuperscript{43}Y. Ganin and V. Lempitsky, Unsupervised Domain Adaptation by Backpropagation, ICML 15.

Image: Courtesy to Y. Ganin.

G. Csurka, DA for Visual Applications
Results on OFF31 (US sse)

- **SMDA** - Stacked Marginalized Denoising Autoencoders, Chen et al. ICML’12.
- **DLID** - Interpolating between Domains Chopra et al. RL-WS ICML13.
- **DAN** - Deep Adaptation Networks, Long et al., CoRR’15.
- **DAB** - Domain Adaptation by Backpropagation, Ganin and Lempitsky, ICML’15.

<table>
<thead>
<tr>
<th></th>
<th>bike</th>
<th>desktop chair</th>
<th>A -&gt; W</th>
<th>D -&gt; W</th>
<th>W -&gt; D</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>dslr</strong></td>
<td></td>
<td></td>
<td>69.8</td>
<td>94.44</td>
<td>97.28</td>
<td>87.17</td>
</tr>
<tr>
<td><strong>amazon</strong></td>
<td></td>
<td></td>
<td>51.9</td>
<td>78.2</td>
<td>89.9</td>
<td>73.33</td>
</tr>
<tr>
<td><strong>webcam</strong></td>
<td></td>
<td></td>
<td>60.5</td>
<td>94.8</td>
<td>98.5</td>
<td>84.6</td>
</tr>
<tr>
<td><strong>SMDA</strong></td>
<td>64.5</td>
<td>95.2</td>
<td>98.6</td>
<td>86.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>DAB</strong></td>
<td>73</td>
<td>96.4</td>
<td>99.2</td>
<td>89.53</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

1. Introduction
   Benchmark Datasets

2. Main domain adaptation methods
   Instance reweighing methods
   Parameter based methods
   Feature transformation-based methods

3. Combined methods
   Joint feature transform and parameter adaptation
   Joint feature/instance selection and feature transform
   Joint instance selection, feature and parameter adaptation

4. Heterogeneous features

5. Multiple sources

6. Deep Learning

7. Conclusion and Perspectives
Conclusion

▶ Many method exploits the Maximum Mean Discrepancy (MMD)

▶ Most popular methods are based on feature transform
  • Manifold based methods (GFS,GFK, BGFK)
  • Unsupervised subspace alignment (SA, LSSA)
  • Stacked marginalized denoising autoencoders (SMDA)

▶ Best performing methods exploit jointly instance selection, feature transform and parameter adaptation
  • Adaptive Transductive Transfer Machines (ATTM)
  • Statistically Invariant Embedding (SIE-CC)

▶ DeCAF features yield to significantly better results
  • Adding adaptation methods can further improve the results.
  • Using Deep Learning to perform adaptation performs the best.
More challenging transfer problems
Continuous adaptation
Adapting object detection

Hoffman et al. LSDA: Large Scale Detection Through Adaptation, NIPS’14.

Image: Courtesy to T. Hoffman.
Adapting video action recognition

- adapting between single and double, between tennis and badminton

---


G. Csurka, DA for Visual Applications
Thank you!