Connecting the Dots with Landmarks: Discriminatively Learning Domain-Invariant Features for Unsupervised Domain Adaptation

Boqing Gong, Kristen Grauman and Fei Sha

(ICML 2013)

Discussion by: Piyush Rai

July 03, 2014
Domain Adaptation

Source Domain (Taken from Amazon)

Target Domain (Taken by webcam)

Unsupervised Domain Adaptation

- Source domain examples are labeled
- Target domain has only unlabeled examples
Identify source domain examples ("landmarks") that appear to be like target domain examples. Move them (without labels) to the target domain.

- Use an existing framework (Geodesic Flow Kernel) to learn domain-invariant features that are good for either domain.

- Repeat (1) and (2) to get multiple sets of domain-invariant features (each using a different set of landmarks) and use their weighted concatenation.
Identify source domain examples ("landmarks") that appear to be like target domain examples. Move them (without labels) to the target domain.

Use an existing framework (Geodesic Flow Kernel) to learn domain-invariant features that are good for either domain.

Repeat (1) and (2) to get multiple sets of domain-invariant features (each using a different set of landmarks) and use their weighted concatenation.
This Paper

1. Identify source domain examples ("landmarks") that appear to be like target domain examples. Move them (without labels) to the target domain.

2. Use an existing framework (Geodesic Flow Kernel) to learn domain-invariant features that are good for either domain.

3. Repeat (1) and (2) to get multiple sets of domain-invariant features (each using a different set of landmarks) and use their weighted concatenation.
Summary of the method

Blue: Source domain examples; Red: Target domain examples

- Each choice ‘q’ of the set of landmarks + Geodesic Flow Kernel on domain pairs after landmark selection ⇒ a domain-invariant feature space $\Phi_q(.)$

- Final step: weighted concatenation of the $\Phi_q$'s
Summary of the method

Blue: Source domain examples; Red: Target domain examples

Each choice ‘q’ of the set of landmarks + Geodesic Flow Kernel on domain pairs after landmark selection ⇒ a domain-invariant feature space \( \Phi_q(.) \)

Final step: weighted concatenation of the \( \Phi_q \)'s
Summary of the method

Blue: Source domain examples; Red: Target domain examples

Each choice ‘q’ of the set of landmarks + Geodesic Flow Kernel on domain pairs after landmark selection \(\Rightarrow\) a domain-invariant feature space \(\Phi_q(.)\)

Final step: weighted concatenation of the \(\Phi_q\)’s
Step 1: Landmark Selection

- **Source:** \( \{(x_m, y_m)\}_{m=1}^{M} \)
- **Target:** \( \{x_n\}_{n=1}^{N} \)

Assume \( D \) dimensional features

Define binary indicator variables \( \alpha = \{\alpha_m\}_{m=1}^{M} \)

If \( \alpha_m = 1 \) then source example \( x_m \) is a landmark

**Goal:** Select landmarks to match selected set and target domain maximally

\[
\min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(x_m) - \frac{1}{N} \sum_n \phi(x_n) \right\|_H^2
\]

s.t.

\[
\frac{1}{\sum_m \alpha_m} \sum_m \alpha_m y_{mc} = \frac{1}{M} \sum_m y_{mc}
\]

Solved using a QP relaxation via transformation \( \beta_m = \alpha_m / (\sum_m \alpha_m) \).
Step 1: Landmark Selection

- Source: \( \{(x_m, y_m)\}_{m=1}^{M} \), Target: \( \{x_n\}_{n=1}^{N} \), assume \( D \) dimensional features

- Define binary indicator variables \( \alpha = \{\alpha_m\}_{m=1}^{M} \)
  - If \( \alpha_m = 1 \) then source example \( x_m \) is a landmark

- Goal: Select landmarks to match selected set and target domain maximally

\[
\min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(x_m) - \frac{1}{N} \sum_n \phi(x_n) \right\|_2^2 \quad \text{s.t.} \quad \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m y_{mc} = \frac{1}{M} \sum_m y_{mc}
\]

- Solved using a QP relaxation via transformation \( \beta_m = \alpha_m / (\sum_m \alpha_m) \)
Step 1: Landmark Selection

- Source: \(\{(x_m, y_m)\}_{m=1}^M\), Target: \(\{x_n\}_{n=1}^N\), assume \(D\) dimensional features

- Define binary indicator variables \(\alpha = \{\alpha_m\}_{m=1}^M\)
  - If \(\alpha_m = 1\) then source example \(x_m\) is a landmark

- Goal: Select landmarks to match selected set and target domain maximally

\[
\min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(x_m) - \frac{1}{N} \sum_n \phi(x_n) \right\|_2^2
\]

s.t.
\[
\frac{1}{\sum_m \alpha_m} \sum_m \alpha_m y_{mc} = \frac{1}{M} \sum_m y_{mc}
\]

- Solved using a QP relaxation via transformation \(\beta_m = \alpha_m / (\sum_m \alpha_m)\)
Step 1: Landmark Selection

- Source: \( \{(x_m, y_m)\}_{m=1}^M \), Target: \( \{x_n\}_{n=1}^N \), assume \( D \) dimensional features

- Define binary indicator variables \( \alpha = \{\alpha_m\}_{m=1}^M \)
  - If \( \alpha_m = 1 \) then source example \( x_m \) is a landmark

- Goal: Select landmarks to match selected set and target domain maximally

\[
\min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(x_m) - \frac{1}{N} \sum_n \phi(x_n) \right\|_2^2 \\
\text{s.t.} \quad \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m y_{mc} = \frac{1}{M} \sum_m y_{mc}
\]

- Solved using a QP relaxation via transformation \( \beta_m = \alpha_m / (\sum_m \alpha_m) \)
Step 1: Landmark Selection

- Source: \(\{(x_m, y_m)\}_{m=1}^M\), Target: \(\{x_n\}_{n=1}^N\), assume \(D\) dimensional features

- Define binary indicator variables \(\alpha = \{\alpha_m\}_{m=1}^M\)
  - If \(\alpha_m = 1\) then source example \(x_m\) is a landmark

- Goal: Select landmarks to match selected set and target domain maximally

\[
\min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(x_m) - \frac{1}{N} \sum_n \phi(x_n) \right\|^2_H
\]

s.t. \[
\frac{1}{\sum_m \alpha_m} \sum_m \alpha_m y_{mc} = \frac{1}{M} \sum_m y_{mc}
\]

- Solved using a QP relaxation via transformation \(\beta_m = \alpha_m / (\sum_m \alpha_m)\)
Step 1: Landmark Selection

- Landmark selection objective function

\[
\min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(x_m) - \frac{1}{N} \sum_n \phi(x_n) \right\|_H^2
\]

- Assume kernel \( K(., .) \) associated with \( \phi(.) \)

\[
K(x_i, x_j) = \exp\{-(x_i - x_j)^\top M (x_i - x_j)/\sigma^2\}
\]

- Different choices of \( \sigma \) will lead to different sets of landmarks
  - Each \( \sigma_q \in \{\sigma_{\text{min}}, \sigma_{\text{max}}\}_{q=1}^Q \) will give landmarks \( L^q = \{(x_m, y_m) : \alpha_m = 1\} \)
  - Different choices of \( \sigma \) explore different granularities in landmark selection
Step 1: Landmark Selection

- Landmark selection objective function

\[
\min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(x_m) - \frac{1}{N} \sum_n \phi(x_n) \right\|_H^2
\]

- Assume kernel \( K(.,.) \) associated with \( \phi(.) \)

\[
K(x_i, x_j) = \exp\left\{ - (x_i - x_j)^\top M (x_i - x_j) / \sigma^2 \right\}
\]

- Different choices of \( \sigma \) will lead to different sets of landmarks
  - Each \( \sigma_q \in \{\sigma_{\min}, \sigma_{\max}\}^{Q}_{q=1} \) will give landmarks \( L^q = \{(x_m, y_m) : \alpha_m = 1\} \)
  - Different choices of \( \sigma \) explore different granularities in landmark selection
Step 1: Landmark Selection

- Landmark selection objective function

\[ \min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(x_m) - \frac{1}{N} \sum_n \phi(x_n) \right\|_H^2 \]

- Assume kernel \( K(., .) \) associated with \( \phi(.) \)

\[ K(x_i, x_j) = \exp\{-(x_i - x_j)^\top M (x_i - x_j)/\sigma^2\} \]

- Different choices of \( \sigma \) will lead to different sets of landmarks

- Each \( \sigma_q \in \{\sigma_{min}, \sigma_{max}\}_{q=1}^Q \) will give landmarks \( L^q = \{ (x_m, y_m) : \alpha_m = 1 \} \)

- Different choices of \( \sigma \) explore different granularities in landmark selection
Step 1: Landmark Selection

- Landmark selection objective function

\[
\min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(x_m) - \frac{1}{N} \sum_n \phi(x_n) \right\|_H^2
\]

- Assume kernel \( K(., .) \) associated with \( \phi(.) \)

\[
K(x_i, x_j) = \exp\left\{ - (x_i - x_j)^\top M (x_i - x_j) / \sigma^2 \right\}
\]

- Different choices of \( \sigma \) will lead to different sets of landmarks
  - Each \( \sigma_q \in \{\sigma_{\text{min}}, \sigma_{\text{max}}\}_{q=1}^Q \) will give landmarks \( \mathcal{L}^q = \{(x_m, y_m) : \alpha_m = 1\} \)

- Different choices of \( \sigma \) explore different granularities in landmark selection
Step 1: Landmark Selection

- Landmark selection objective function

\[
\min_{\alpha} \left\| \frac{1}{\sum_m \alpha_m} \sum_m \alpha_m \phi(x_m) - \frac{1}{N} \sum_n \phi(x_n) \right\|_H^2
\]

- Assume kernel \( K(., .) \) associated with \( \phi(.) \)

\[
K(x_i, x_j) = \exp \left\{ - (x_i - x_j)^\top M (x_i - x_j) / \sigma^2 \right\}
\]

- Different choices of \( \sigma \) will lead to different sets of landmarks
  - Each \( \sigma_q \in \{\sigma_{min}, \sigma_{max}\}_{q=1}^Q \) will give landmarks \( L^q = \{(x_m, y_m) : \alpha_m = 1\} \)
  - Different choices of \( \sigma \) explore different granularities in landmark selection
Step 2: Learning Domain-Invariant Features

- Move landmark points (without labels) from source to target

- New source: $D^q_S = D_S \setminus L^q$, New target: $D^q_T = D_T \cup L^q$

- Let $P_S, P_T \in \mathbb{R}^{D \times d}$ be the PCA basis of source and target, respectively

- Geodesic flow $\{\Phi(t) : t \in [0, 1]\}$ between $P_S$ and $P_T$ is a path connecting the two domains: $P_S = \Phi(0), P_T = \Phi(1)$

- Projection of a point $x$ along the geodesic
  
  $z^\infty = \{\Phi(t)^T x : t \in [0, 1]\}$

- Domain-invariant $z^\infty$ defines infinite many features smoothly varying as being source-like to target-like
Step 2: Learning Domain-Invariant Features

- Move landmark points (without labels) from source to target

- New source: $\mathcal{D}_S^q = \mathcal{D}_S \setminus \mathcal{L}_q$, New target: $\mathcal{D}_T^q = \mathcal{D}_T \cup \mathcal{L}_q$

- Let $P_S, P_T \in \mathbb{R}^{D \times d}$ be the PCA basis of source and target, respectively

- Geodesic flow $\{\Phi(t) : t \in [0, 1]\}$ between $P_S$ and $P_T$ is a path connecting the two domains: $P_S = \Phi(0), P_T = \Phi(1)$

- Projection of a point $x$ along the geodesic

  $$z^\infty = \{\Phi(t)^\top x : t \in [0, 1]\}$$

- Domain-invariant $z^\infty$ defines infinite many features smoothly varying as being source-like to target-like
Step 2: Learning Domain-Invariant Features

- Move landmark points (without labels) from source to target

- New source: $D_S^q = D_S \setminus L^q$, New target: $D_T^q = D_T \cup L^q$

- Let $P_S, P_T \in \mathbb{R}^{D \times d}$ be the PCA basis of source and target, respectively

- Geodesic flow $\{\Phi(t) : t \in [0, 1]\}$ between $P_S$ and $P_T$ is a path connecting the two domains: $P_S = \Phi(0), P_T = \Phi(1)$

- Projection of a point $x$ along the geodesic

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

- Domain-invariant $z^\infty$ defines infinite many features smoothly varying as being source-like to target-like
Step 2: Learning Domain-Invariant Features

- Move landmark points (without labels) from source to target

- New source: $\mathcal{D}_S^q = \mathcal{D}_S \setminus \mathcal{L}^q$, New target: $\mathcal{D}_T^q = \mathcal{D}_T \cup \mathcal{L}^q$

- Let $P_S, P_T \in \mathbb{R}^{D \times d}$ be the PCA basis of source and target, respectively

- Geodesic flow $\{\Phi(t) : t \in [0,1]\}$ between $P_S$ and $P_T$ is a path connecting the two domains: $P_S = \Phi(0), P_T = \Phi(1)$

- Projection of a point $x$ along the geodesic

$$z^\infty = \{\Phi(t)^T x : t \in [0,1]\}$$

- Domain-invariant $z^\infty$ defines infinite many features smoothly varying as being source-like to target-like
Step 2: Learning Domain-Invariant Features

- Move landmark points (without labels) from source to target

- New source: $D^q_S = D_S \setminus L^q$, New target: $D^q_T = D_T \cup L^q$

- Let $P_S, P_T \in \mathbb{R}^{D \times d}$ be the PCA basis of source and target, respectively

- Geodesic flow $\{\Phi(t) : t \in [0, 1]\}$ between $P_S$ and $P_T$ is a path connecting the two domains: $P_S = \Phi(0), P_T = \Phi(1)$

- Projection of a point $x$ along the geodesic

  $$z^\infty = \{\Phi(t)^T x : t \in [0, 1]\}$$

- Domain-invariant $z^\infty$ defines infinite many features smoothly varying as being source-like to target-like
Step 2: Learning Domain-Invariant Features

- Move landmark points (without labels) from source to target

New source: $\mathcal{D}_S^q = \mathcal{D}_S \setminus \mathcal{L}^q$, New target: $\mathcal{D}_T^q = \mathcal{D}_T \cup \mathcal{L}^q$

- Let $P_S, P_T \in \mathbb{R}^{D \times d}$ be the PCA basis of source and target, respectively

- Geodesic flow $\{\Phi(t) : t \in [0, 1]\}$ between $P_S$ and $P_T$ is a path connecting the two domains: $P_S = \Phi(0)$, $P_T = \Phi(1)$

- Projection of a point $x$ along the geodesic

$$z^\infty = \{\Phi(t)^\top x : t \in [0, 1]\}$$

- Domain-invariant $z^\infty$ defines infinite many features smoothly varying as being source-like to target-like
Geodesic Flow Kernel

- Similarity between any two examples (same or different domains)

\[ K_q(x_i, x_j) = \langle z_i^\infty, z_j^\infty \rangle = x_i^\top \int_0^1 \Phi(t)\Phi(t)^\top dt \ x_j = x_i^\top G_q \ x_j \]

- Matrix \( G_q \) can be computed using SVD of \( P_S^\top P_T \) and \( R_S^\top P_T \) (where \( R_S \) is the orthogonal complement to \( P_S \); details in another paper: Geodesic Flow Kernels for Unsupervised Domain Adaptation, Gong et al., CVPR 2012)

- Domain-invariant feature vector for an example \( x \): \( \Phi_q(x) = \sqrt{G_q}x \)

- There will be \( Q \) such feature vectors \( \{\Phi_q(x)\}_{q=1}^Q \) (one for each set of landmarks)

- How to combine \( \{\Phi_q(x)\}_{q=1}^Q \) to get a single feature vector?
Geodesic Flow Kernel

- Similarity between any two examples (same or different domains)

\[ K_q(x_i, x_j) = \langle z_i^\infty, z_j^\infty \rangle = x_i^\top \int_0^1 \Phi(t)\Phi(t)^\top dt \ x_j = x_i^\top G_q \ x_j \]

- Matrix \( G_q \) can be computed using SVD of \( P_S^\top P_T \) and \( R_S^\top P_T \) (where \( R_S \) is the orthogonal complement to \( P_S \); details in another paper: Geodesic Flow Kernels for Unsupervised Domain Adaptation, Gong et al., CVPR 2012)

- Domain-invariant feature vector for an example \( x \): \( \Phi_q(x) = \sqrt{G_q}x \)

- There will be \( Q \) such feature vectors \( \{\Phi_q(x)\}_{q=1}^Q \) (one for each set of landmarks)

- How to combine \( \{\Phi_q(x)\}_{q=1}^Q \) to get a single feature vector?
Geodesic Flow Kernel

- Similarity between any two examples (same or different domains)
  
  \[ K_q(x_i, x_j) = \langle z_i^\infty, z_j^\infty \rangle = x_i^\top \int_0^1 \Phi(t)\Phi(t)^\top dt \ x_j = x_i^\top G_q \ x_j \]

- Matrix \( G_q \) can be computed using SVD of \( P_S^\top P_T \) and \( R_S^\top P_T \) (where \( R_S \) is the orthogonal complement to \( P_S \); details in another paper: Geodesic Flow Kernels for Unsupervised Domain Adaptation, Gong et al., CVPR 2012)

- Domain-invariant feature vector for an example \( x \): \( \Phi_q(x) = \sqrt{G_q}x \)

- There will be \( Q \) such feature vectors \( \{ \Phi_q(x) \}_{q=1}^Q \) (one for each set of landmarks)

- How to combine \( \{ \Phi_q(x) \}_{q=1}^Q \) to get a single feature vector?
Geodesic Flow Kernel

- Similarity between any two examples (same or different domains)

\[ K_q(x_i, x_j) = \langle z_i^\infty, z_j^\infty \rangle = x_i^\top \int_0^1 \Phi(t)\Phi(t)^\top dt \ x_j = x_i^\top G_q x_j \]

- Matrix $G_q$ can be computed using SVD of $P_S^\top P_T$ and $R_S^\top P_T$ (where $R_S$ is the orthogonal complement to $P_S$; details in another paper: Geodesic Flow Kernels for Unsupervised Domain Adaptation, Gong et al., CVPR 2012)

- Domain-invariant feature vector for an example $x$: $\Phi_q(x) = \sqrt{G_q}x$

- There will be $Q$ such feature vectors $\{\Phi_q(x)\}_{q=1}^Q$ (one for each set of landmarks)

- How to combine $\{\Phi_q(x)\}_{q=1}^Q$ to get a single feature vector?
Geodesic Flow Kernel

- Similarity between any two examples (same or different domains)

\[ K_q(x_i, x_j) = \langle z_i^\infty, z_j^\infty \rangle = x_i^\top \int_0^1 \Phi(t)\Phi(t)^\top dt \ x_j = x_i^\top G_q \ x_j \]

- Matrix \( G_q \) can be computed using SVD of \( P_S^\top P_T \) and \( R_S^\top P_T \) (where \( R_S \) is the orthogonal complement to \( P_S \); details in another paper: Geodesic Flow Kernels for Unsupervised Domain Adaptation, Gong et al., CVPR 2012)

- Domain-invariant feature vector for an example \( x \): \( \Phi_q(x) = \sqrt{G_q} x \)

- There will be \( Q \) such feature vectors \( \{\Phi_q(x)\}_{q=1}^Q \) (one for each set of landmarks)

- How to combine \( \{\Phi_q(x)\}_{q=1}^Q \) to get a single feature vector?
Step 3: Combining domain-invariant feature vectors

- (Weighted) concatenation of the feature vectors \(\{\sqrt{w_q} \Phi_q(x)\}\)\(^Q\)

- Solve a multiple kernel learning problem to learn the weights \(\{w_q\}\)\(^Q\)

- Labeled training data \(\mathcal{D}_{TRAIN} = \bigcup_q \mathcal{L}^q\) and kernel defined as

\[
F = \sum_q w_q G_q, \quad \text{s.t.} \quad w_q \geq 0 \quad \text{and} \quad \sum_q w_q = 1
\]

- Validation set \(\mathcal{D}_{VAL} = \mathcal{D}_S \setminus \mathcal{D}_{TRAIN}\) to select the optimal \(\{w_q\}\)\(^Q\)
Step 3: Combining domain-invariant feature vectors

- (Weighted) concatenation of the feature vectors \( \{\sqrt{w_q} \Phi_q(x)\}^{Q}_{q=1} \)

- Solve a multiple kernel learning problem to learn the weights \( \{w_q\}^{Q}_{q=1} \)

- Labeled training data \( \mathcal{D}_{TRAIN} = \bigcup_q \mathcal{L}^q \) and kernel defined as

\[
F = \sum_q w_q G_q, \quad \text{s.t.} \quad w_q \geq 0 \quad \text{and} \quad \sum_q w_q = 1
\]

- Validation set \( \mathcal{D}_{VAL} = \mathcal{D}_S \setminus \mathcal{D}_{TRAIN} \) to select the optimal \( \{w_q\}^{Q}_{q=1} \)
(Weighted) concatenation of the feature vectors \( \{ \sqrt{w_q} \Phi_q(x) \}^Q_{q=1} \)

Solve a multiple kernel learning problem to learn the weights \( \{w_q\}^Q_{q=1} \)

Labeled training data \( \mathcal{D}_{TRAIN} = \bigcup_q \mathcal{L}^q \) and kernel defined as

\[
F = \sum_q w_q G_q, \quad \text{s.t.} \quad w_q \geq 0 \quad \text{and} \quad \sum_q w_q = 1
\]

Validation set \( \mathcal{D}_{VAL} = \mathcal{D}_S \setminus \mathcal{D}_{TRAIN} \) to select the optimal \( \{w_q\}^Q_{q=1} \)
Step 3: Combining domain-invariant feature vectors

- (Weighted) concatenation of the feature vectors \( \{ \sqrt{w_q} \Phi_q(x) \}^Q_{q=1} \)

- Solve a multiple kernel learning problem to learn the weights \( \{w_q\}^Q_{q=1} \)

- Labeled training data \( \mathcal{D}_{TRAIN} = \bigcup_q \mathcal{L}^q \) and kernel defined as

\[
F = \sum_q w_q G_q, \quad \text{s.t.} \quad w_q \geq 0 \quad \text{and} \quad \sum_q w_q = 1
\]

- Validation set \( \mathcal{D}_{VAL} = \mathcal{D}_S \setminus \mathcal{D}_{TRAIN} \) to select the optimal \( \{w_q\}^Q_{q=1} \)
Experiments

- Object detection on data from 4 domains
- Sentiment analysis on data from 4 domains
- Baselines
  - TCA: Transfer Component Analysis
  - KMM: Kernel Mean Matching
  - GFS: Geodesic Flow Sampling
  - GFK: Geodesic Flow Kernel
  - SCL: Structured Correspondence Learning
  - Metric: Learns similarities between source and target examples
Experiments: Object Detection

- Used 4 domains: CALTECH, AMAZON, WEBCAM, DSLR
- $\sigma_q = 2^q \sigma_0$, $q = \{-6, -5, \ldots, 5, 6\}$, $\sigma_0$: median pairwise distance

Table 1. Recognition accuracies on 9 pairs of source/target domains are reported. C: caltech, A: amazon, W: webcam, D: dslr. The proposed method (LANDMARK) performs the best on 8 out of 9 pairs, among all unsupervised methods.

<table>
<thead>
<tr>
<th>%</th>
<th>A→C</th>
<th>A→D</th>
<th>A→W</th>
<th>C→A</th>
<th>C→D</th>
<th>C→W</th>
<th>W→A</th>
<th>W→C</th>
<th>W→D</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO ADAPTATION</td>
<td>41.7</td>
<td>41.4</td>
<td>34.2</td>
<td>51.8</td>
<td>54.1</td>
<td>46.8</td>
<td>31.1</td>
<td>31.5</td>
<td>70.7</td>
</tr>
<tr>
<td>TCA (Pan et al., 2009)</td>
<td>35.0</td>
<td>36.3</td>
<td>27.8</td>
<td>41.4</td>
<td>45.2</td>
<td>32.5</td>
<td>24.2</td>
<td>22.5</td>
<td>80.2</td>
</tr>
<tr>
<td>GFS (Gopalan et al., 2011)</td>
<td>39.2</td>
<td>36.3</td>
<td>33.6</td>
<td>43.6</td>
<td>40.8</td>
<td>36.3</td>
<td>33.5</td>
<td>30.9</td>
<td>75.7</td>
</tr>
<tr>
<td>GFK (Gong et al., 2012)</td>
<td>42.2</td>
<td>42.7</td>
<td>40.7</td>
<td>44.5</td>
<td>43.3</td>
<td>44.7</td>
<td>31.8</td>
<td>30.8</td>
<td>75.6</td>
</tr>
<tr>
<td>SCL (Blitzer et al., 2006)</td>
<td>42.3</td>
<td>36.9</td>
<td>34.9</td>
<td>49.3</td>
<td>42.0</td>
<td>39.3</td>
<td>34.7</td>
<td>32.5</td>
<td>83.4</td>
</tr>
<tr>
<td>KMM (Huang et al., 2007)</td>
<td>42.2</td>
<td>42.7</td>
<td>42.4</td>
<td>48.3</td>
<td>53.5</td>
<td>45.8</td>
<td>31.9</td>
<td>29.0</td>
<td>72.0</td>
</tr>
<tr>
<td>METRIC (Saenko et al., 2010)</td>
<td>42.4</td>
<td>42.9</td>
<td>49.8</td>
<td>46.6</td>
<td>47.6</td>
<td>42.8</td>
<td>38.6</td>
<td>33.0</td>
<td>87.1</td>
</tr>
<tr>
<td>LANDMARK (ours)</td>
<td>45.5</td>
<td>47.1</td>
<td>46.1</td>
<td>56.7</td>
<td>57.3</td>
<td>49.5</td>
<td>40.2</td>
<td>35.4</td>
<td>75.2</td>
</tr>
</tbody>
</table>

Table 2. Contrasting LANDMARK to several variants, illustrating the importance of our landmark selection algorithm.

<table>
<thead>
<tr>
<th>%</th>
<th>A→C</th>
<th>A→D</th>
<th>A→W</th>
<th>C→A</th>
<th>C→D</th>
<th>C→W</th>
<th>W→A</th>
<th>W→C</th>
<th>W→D</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANDMARK (ours)</td>
<td>45.5</td>
<td>47.1</td>
<td>46.1</td>
<td>56.7</td>
<td>57.3</td>
<td>49.5</td>
<td>40.2</td>
<td>35.4</td>
<td>75.2</td>
</tr>
<tr>
<td>Rand. Sel.</td>
<td>44.5</td>
<td>44.5</td>
<td>41.9</td>
<td>53.8</td>
<td>49.9</td>
<td>49.5</td>
<td>39.8</td>
<td>34.1</td>
<td>74.2</td>
</tr>
<tr>
<td>SWAP</td>
<td>41.3</td>
<td>47.8</td>
<td>37.6</td>
<td>46.2</td>
<td>42.0</td>
<td>46.1</td>
<td>38.2</td>
<td>32.2</td>
<td>70.1</td>
</tr>
<tr>
<td>UNBALANCED</td>
<td>37.0</td>
<td>36.9</td>
<td>38.3</td>
<td>55.3</td>
<td>49.0</td>
<td>50.1</td>
<td>39.4</td>
<td>34.9</td>
<td>73.9</td>
</tr>
<tr>
<td>EUC. Sel.</td>
<td>44.5</td>
<td>44.0</td>
<td>41.0</td>
<td>50.2</td>
<td>40.1</td>
<td>45.1</td>
<td>39.1</td>
<td>34.5</td>
<td>67.5</td>
</tr>
</tbody>
</table>
### Effect of scale on landmark selection

<table>
<thead>
<tr>
<th>HEADPHONE from WEBCAM</th>
<th>Landmarks at scale $\sigma = 2^6 \sigma_0$</th>
<th>Landmarks at scale $\sigma = 2^3 \sigma_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Headphones" /></td>
<td><img src="image2" alt="Landmarks" /></td>
<td><img src="image3" alt="Landmarks" /></td>
</tr>
<tr>
<td>Landmarks at scale $\sigma = 2^0 \sigma_0$</td>
<td>Landmarks at scale $\sigma = 2^{-3} \sigma_0$</td>
<td>Examples of non-landmarks</td>
</tr>
<tr>
<td><img src="image1" alt="Headphones" /></td>
<td><img src="image2" alt="Landmarks" /></td>
<td><img src="image3" alt="Landmarks" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MUG from WEBCAM</th>
<th>Landmarks at scale $\sigma = 2^6 \sigma_0$</th>
<th>Landmarks at scale $\sigma = 2^3 \sigma_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4" alt="Mugs" /></td>
<td><img src="image5" alt="Landmarks" /></td>
<td><img src="image6" alt="Landmarks" /></td>
</tr>
<tr>
<td>Landmarks at scale $\sigma = 2^0 \sigma_0$</td>
<td>Landmarks at scale $\sigma = 2^{-3} \sigma_0$</td>
<td>Examples of non-landmarks</td>
</tr>
<tr>
<td><img src="image4" alt="Mugs" /></td>
<td><img src="image5" alt="Landmarks" /></td>
<td><img src="image6" alt="Landmarks" /></td>
</tr>
</tbody>
</table>
Amazon product reviews from 4 domains: Appliances, DVDs, Books, Electronics

Table 3. Sentiment classification accuracies on target domains. K: kitchen, D: dvd, B: books, E: electronics

<table>
<thead>
<tr>
<th></th>
<th>K→D</th>
<th>D→B</th>
<th>B→E</th>
<th>E→K</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO ADAPTATION</td>
<td>72.7</td>
<td>73.4</td>
<td>73.0</td>
<td>81.4</td>
</tr>
<tr>
<td>TCA</td>
<td>60.4</td>
<td>61.4</td>
<td>61.3</td>
<td>68.7</td>
</tr>
<tr>
<td>GFS</td>
<td>67.9</td>
<td>68.6</td>
<td>66.9</td>
<td>75.1</td>
</tr>
<tr>
<td>GFK</td>
<td>69.0</td>
<td>71.3</td>
<td>68.4</td>
<td>78.2</td>
</tr>
<tr>
<td>SCL</td>
<td>72.8</td>
<td>76.2</td>
<td>75.0</td>
<td>82.9</td>
</tr>
<tr>
<td>KMM</td>
<td>72.2</td>
<td>78.6</td>
<td>76.9</td>
<td>83.5</td>
</tr>
<tr>
<td>METRIC</td>
<td>70.6</td>
<td>72.0</td>
<td>72.2</td>
<td>77.1</td>
</tr>
<tr>
<td>LANDMARK (ours)</td>
<td><strong>75.1</strong></td>
<td><strong>79.0</strong></td>
<td><strong>78.5</strong></td>
<td><strong>83.4</strong></td>
</tr>
</tbody>
</table>
Other alternatives for landmark section could be considered, e.g.,

- Learn a classifier to separate source and target domain. The misclassified source examples could be treated as “target-like”

Other follow-up works:

- Learning a shared projection $W \in \mathbb{R}^{D \times d}$ for source and target to minimize the Maximum Mean Discrepancy (Baktashmotlagh et al, ICCV 2013)

- Minimizing the Hellinger Distance between source and target (Baktashmotlagh et al, CVPR 2014), e.g., by
  - Selecting a subset of source examples to minimize HD
  - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ to minimize HD
Discussion

- Other alternatives for landmark section could be considered, e.g.,
  - Learn a classifier to separate source and target domain. The misclassified source examples could be treated as “target-like”

- Other follow-up works:
  - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ for source and target to minimize the Maximum Mean Discrepancy (Baktashmotlagh et al, ICCV 2013)
  - Minimizing the Hellinger Distance between source and target (Baktashmotlagh et al, CVPR 2014), e.g., by
    - Selecting a subset of source examples to minimize HD
    - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ to minimize HD
Discussion

- Other alternatives for landmark section could be considered, e.g.,
  - Learn a classifier to separate source and target domain. The misclassified source examples could be treated as “target-like”

- Other follow-up works:
  - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ for source and target to minimize the Maximum Mean Discrepancy (Baktashmotlagh et al, ICCV 2013)
  - Minimizing the Hellinger Distance between source and target (Baktashmotlagh et al, CVPR 2014), e.g., by
    - Selecting a subset of source examples to minimize HD
    - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ to minimize HD
Discussion

- Other alternatives for landmark section could be considered, e.g.,
  - Learn a classifier to separate source and target domain. The misclassified source examples could be treated as “target-like”

- Other follow-up works:
  - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ for source and target to minimize the Maximum Mean Discrepancy (Baktashmotlagh et al, ICCV 2013)
    - Minimizing the Hellinger Distance between source and target (Baktashmotlagh et al, CVPR 2014), e.g., by
      - Selecting a subset of source examples to minimize HD
      - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ to minimize HD
Discussion

- Other alternatives for landmark section could be considered, e.g.,
  - Learn a classifier to separate source and target domain. The misclassified source examples could be treated as “target-like”

- Other follow-up works:
  - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ for source and target to minimize the Maximum Mean Discrepancy (Baktashmotlagh et al, ICCV 2013)
  - Minimizing the Hellinger Distance between source and target (Baktashmotlagh et al, CVPR 2014), e.g., by
    - Selecting a subset of source examples to minimize HD
    - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ to minimize HD
Other alternatives for landmark section could be considered, e.g.,

- Learn a classifier to separate source and target domain. The misclassified source examples could be treated as “target-like”

Other follow-up works:

- Learning a shared projection $W \in \mathbb{R}^{D \times d}$ for source and target to minimize the Maximum Mean Discrepancy (Baktashmotlagh et al, ICCV 2013)

- Minimizing the Hellinger Distance between source and target (Baktashmotlagh et al, CVPR 2014), e.g., by
  - Selecting a subset of source examples to minimize HD
  - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ to minimize HD
Discussion

- Other alternatives for landmark section could be considered, e.g.,
  - Learn a classifier to separate source and target domain. The misclassified source examples could be treated as “target-like”

- Other follow-up works:
  - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ for source and target to minimize the Maximum Mean Discrepancy (Baktashmotlagh et al, ICCV 2013)

  - Minimizing the Hellinger Distance between source and target (Baktashmotlagh et al, CVPR 2014), e.g., by
    - Selecting a subset of source examples to minimize HD
    - Learning a shared projection $W \in \mathbb{R}^{D \times d}$ to minimize HD
Thanks! Questions?