• Suppose we have access to only samples from two distributions $X \sim P_A$ and $Y \sim P_B$.

• Can we tell if $P_A = P_B$?
  – Two-sample test problem
Suppose we have access to only samples from two distributions $X \sim P_A$ and $Y \sim P_B$.

Can we tell if $P_A = P_B$?

- Two-sample test problem

Maximum Mean Discrepancy [Gretton et al. 2006] is among the best performing measure of discrepancy between distributions for two-sample test.
MMD

$$\left\| \frac{1}{N} \sum_{n=1}^{N} \phi(X_n) - \frac{1}{M} \sum_{m=1}^{M} \phi(Y_m) \right\|^2$$

$$= \frac{1}{N^2} \sum_{n=1}^{N} \sum_{n'=1}^{N} \phi(X_n)^\top \phi(X_{n'}) + \frac{1}{M^2} \sum_{m=1}^{M} \sum_{m'=1}^{M} \phi(Y_m)^\top \phi(Y_{m'}) - \frac{2}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} \phi(X_n)^\top \phi(Y_m)$$

$$= \frac{1}{N^2} \sum_{n=1}^{N} \sum_{n'=1}^{N} k(X_n, X_{n'}) + \frac{1}{M^2} \sum_{m=1}^{M} \sum_{m'=1}^{M} k(Y_m, Y_{m'}) - \frac{2}{MN} \sum_{n=1}^{N} \sum_{m=1}^{N} k(X_n, Y_m)$$

- $\{X_n\} \sim P_A$, $\{Y_m\} \sim P_B$
- $\phi$: feature map
- $k$: universal kernel
What can we use it for?

The opposite direction: learning to make two distributions indistinguishable $\Rightarrow$ small MMD!
What can we use it for?

The opposite direction: learning to make two distributions indistinguishable $\Rightarrow$ small MMD!

Natural fit: domain adaptation

- Make feature representations for source and target domain data indistinguishable
Domain Adaptation/Transfer Learning with MMD

• Correcting Sample Selection Bias by Unlabeled Data [Huang et al. NIPS 2006]
• Transfer Learning via Dimensionality Reduction [Pan et al. AAAI 2008]
• Domain Adaptation via Transfer Component Analysis [Pan et al. IJCAI 2009]
• Connecting the Dots with Landmarks: Discriminatively Learning Domain-Invariant Features for Unsupervised Domain Adaptation [Gong et al. ICML 2013]
• Reshaping Visual Datasets for Domain Adaptation [Gong et al. NIPS 2013]
• Transfer Feature Learning with Joint Distribution Adaptation [Long et al. ICCV 2013]
• Unsupervised Domain Adaptation by Domain Invariant Projection [Baktashmotlagh, ICCV 2013]

• Many more...

• Flexible Transfer Learning under Support and Model Shift [Wang and Schneider, This workshop]
Domain Adaptation

Sentiment classification

- Product reviews (text, tf-idf on words & bigrams)
- Labeled data from source domain, unlabeled data from target domain

\[
\text{Loss} = \text{Loss}_{\text{class}} + \lambda \text{MMD}
\]
## Domain Adaptation

<table>
<thead>
<tr>
<th></th>
<th>D→B</th>
<th>E→B</th>
<th>K→B</th>
<th>B→D</th>
<th>E→D</th>
<th>K→D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SVM</td>
<td>78.3 ± 1.4</td>
<td>71.0 ± 2.0</td>
<td>72.9 ± 2.4</td>
<td>79.0 ± 1.9</td>
<td>72.5 ± 2.9</td>
<td>73.6 ± 1.5</td>
</tr>
<tr>
<td>RBF SVM</td>
<td>77.7 ± 1.2</td>
<td>68.0 ± 1.9</td>
<td>73.2 ± 2.4</td>
<td>79.1 ± 2.3</td>
<td>70.7 ± 1.8</td>
<td>73.0 ± 1.6</td>
</tr>
<tr>
<td>TCA</td>
<td>77.5 ± 1.3</td>
<td>71.8 ± 1.4</td>
<td>68.8 ± 2.4</td>
<td>76.9 ± 1.4</td>
<td>72.5 ± 1.9</td>
<td>73.3 ± 2.4</td>
</tr>
<tr>
<td>NN</td>
<td>76.6 ± 1.8</td>
<td>70.0 ± 2.4</td>
<td>72.8 ± 1.5</td>
<td>78.3 ± 1.6</td>
<td>71.7 ± 2.7</td>
<td>72.7 ± 1.6</td>
</tr>
<tr>
<td>NN MMD*</td>
<td>76.5 ± 2.5</td>
<td>71.8 ± 2.1</td>
<td>72.8 ± 2.4</td>
<td>77.4 ± 2.4</td>
<td>74.3 ± 1.7</td>
<td>73.9 ± 2.4</td>
</tr>
<tr>
<td>NN MMD</td>
<td><strong>78.5 ± 1.5</strong></td>
<td><strong>73.7 ± 2.0</strong></td>
<td><strong>75.7 ± 2.3</strong></td>
<td><strong>79.2 ± 1.7</strong></td>
<td><strong>75.3 ± 2.1</strong></td>
<td><strong>75.0 ± 1.0</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>B→E</th>
<th>D→E</th>
<th>K→E</th>
<th>B→K</th>
<th>D→K</th>
<th>E→K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SVM</td>
<td>72.4 ± 3.0</td>
<td>74.2 ± 1.4</td>
<td>82.7 ± 1.3</td>
<td>75.9 ± 1.8</td>
<td>77.0 ± 1.8</td>
<td>84.5 ± 1.0</td>
</tr>
<tr>
<td>RBF SVM</td>
<td>72.8 ± 2.5</td>
<td>76.3 ± 2.2</td>
<td>82.5 ± 1.4</td>
<td>75.8 ± 2.1</td>
<td>76.0 ± 2.2</td>
<td>82.0 ± 1.4</td>
</tr>
<tr>
<td>TCA</td>
<td>72.1 ± 2.6</td>
<td>75.9 ± 2.7</td>
<td>79.8 ± 1.4</td>
<td>76.8 ± 2.1</td>
<td>76.4 ± 1.7</td>
<td>80.2 ± 1.4</td>
</tr>
<tr>
<td>NN</td>
<td>70.1 ± 3.1</td>
<td>72.8 ± 2.4</td>
<td>82.3 ± 1.0</td>
<td>74.1 ± 1.6</td>
<td>75.8 ± 1.8</td>
<td>84.0 ± 1.5</td>
</tr>
<tr>
<td>NN MMD*</td>
<td>75.6 ± 2.9</td>
<td>78.4 ± 1.6</td>
<td>83.0 ± 1.2</td>
<td>77.9 ± 1.6</td>
<td>78.0 ± 1.9</td>
<td>84.7 ± 1.6</td>
</tr>
<tr>
<td>NN MMD</td>
<td><strong>76.8 ± 2.0</strong></td>
<td><strong>79.1 ± 1.6</strong></td>
<td><strong>83.9 ± 1.0</strong></td>
<td><strong>78.3 ± 1.4</strong></td>
<td><strong>78.6 ± 2.6</strong></td>
<td><strong>85.2 ± 1.1</strong></td>
</tr>
</tbody>
</table>

- 4 domains:
  - D: dvd, B: books, E: electronics, K: kitchen products
- NN MMD*: not-weighted word count feature, weaker than tf-idf
Learning Invariant Features

- If we have labeled data from all domains, factoring out unwanted domain bias still leads to better generalization.

- In general, we can use MMD to make the learned representations invariant to unwanted transformation / variation / bias.
Learning Invariant Features

- Face identification under different lighting

Classification Loss

\[
\sum_{s=1}^{S} \left\| \frac{1}{N_s} \sum_{i:d_i=s} \phi(h_i) - \frac{1}{N} \sum_{n} \phi(h_n) \right\|^2
\]

Multiple lighting conditions: Matching each to the mean
Learning Invariant Features

- Without MMD, test accuracy 72%
  - PCA projection of 2\textsuperscript{nd} hidden layer

Projection of training data (100\% accuracy)
Digits: person identity index, color: lighting condition
Learning Invariant Features

- With MMD, test accuracy 82%
  - PCA projection of 2nd hidden layer

Projection of training data (100% accuracy)
Digits: person identity index, color: lighting condition
Noise-Insensitive Auto-Encoders

• Make auto-encoders robust to noise
  – Push hidden representation for noisy data close to that of clean data with MMD regularizer

Reconstruction Loss

- Small corruption + linear kernel recovers contractive auto-encoder (CAE)
- But we can use more powerful kernels!
Noise-Insensitive Auto-Encoders

- MMD with Gaussian kernel is less sensitive to noise than with linear kernel (CAE).
  - SVM trained to distinguish representation for noisy data from clean data

<table>
<thead>
<tr>
<th>Model</th>
<th>AE</th>
<th>DAE</th>
<th>CAE</th>
<th>MMD</th>
<th>MMD+DAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM Accuracy</td>
<td>78.6</td>
<td>82.5</td>
<td>77.9</td>
<td>61.1</td>
<td>72.9</td>
</tr>
</tbody>
</table>
Learning Deep Generative Models

- Make model samples close to data samples

- Uniform Prior
    - Follow up work from deep learning workshop [Mirza and Osindero] and this workshop [Ajakan et al.].
    - All based on training with adversaries

- Related early work from [MacKay 1995, 1996], [Magdon-Ismail and Atiya, 1998]
• Direct backpropagation through MMD, no adversary required!

Independent Samples

Morphing between two samples

Model trained on MNIST
Learning Deep Generative Models

- Direct backpropagation through MMD, no adversary required!

Independent Samples

Morphing between two samples

Model trained on Frey Face dataset
Q & A

Learning Unbiased Features

Yujia Li, Kevin Swersky and Rich Zemel

University of Toronto
Canadian Institute for Advanced Research