Domain Adaptation in Real-world Applications

Sinno Jialin Pan (Ph.D.)
Scientist@Data Analytics Department, Institute for Infocomm Research (I2R), A*STAR, Singapore
Outline

• **An overview of transfer learning – Part I (Sinno)**
  – Definition of transfer learning (*use the term “domain adaptation” interchangeably in this tutorial*)
  – Summarization of transfer learning settings and approaches

• **Advanced research in transfer learning – Part II (Ivor)**
  – Transfer learning from multiple source domains
  – Optimizing performance measure in transfer learning
  – Transfer knowledge across heterogeneous domains

• **Applications to CV, NLP, WSN (in both Part I & II)**
Transfer of Learning
A psychological point of view

• The study of dependency of human conduct, learning or performance on prior experience.

  - [Thorndike and Woodworth, 1901] explored how individuals would transfer in one context to another context that share similar characteristics.

  ➢ C++ → Java
  ➢ Maths/Physics → Computer Science/Economics
Transfer Learning

In the machine learning community

• The ability of a system to recognize and apply knowledge and skills learned in previous domains/tasks to novel tasks/domains, which share some commonality.

• Given a target domain/task, how to identify the commonality between the domain/task and previous domains/tasks, and transfer knowledge from the previous domains/tasks to the target one?
Transfer Learning

Traditional Machine Learning

Transfer Learning

<table>
<thead>
<tr>
<th>Domain A</th>
<th>Domain B</th>
<th>Domain C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training domains</td>
<td>Training items</td>
<td>Test domains</td>
</tr>
</tbody>
</table>
Transfer Learning
Different fields

- Transfer learning for reinforcement learning.

- Transfer learning for classification, and regression problems.
  [Pan and Yang, A Survey on Transfer Learning, IEEE TKDE 2010]
Motivating Example I: Indoor WiFi localization
Indoor WiFi Localization (cont.)

Training
S=(-37dbm, .., -77dbm), L=(1, 3)
S=(-41dbm, .., -83dbm), L=(1, 4)
...
S=(-49dbm, .., -34dbm), L=(9, 10)
S=(-61dbm, .., -28dbm), L=(15, 22)

Time Period A

Localization model

Test
S=(-37dbm, .., -77dbm)
S=(-41dbm, .., -83dbm)
...
S=(-49dbm, .., -34dbm)
S=(-61dbm, .., -28dbm)

Average Error Distance

~1.5 meters

Drop!

Time Period A

Training
S=(-37dbm, .., -77dbm), L=(1, 3)
S=(-41dbm, .., -83dbm), L=(1, 4)
...
S=(-49dbm, .., -34dbm), L=(9, 10)
S=(-61dbm, .., -28dbm), L=(15, 22)

Time Period B

Localization model

Test
S=(-37dbm, .., -77dbm)
S=(-41dbm, .., -83dbm)
...
S=(-49dbm, .., -34dbm)
S=(-61dbm, .., -28dbm)

~6 meters

Time Period A
Indoor WiFi Localization (cont.)

Training
- Device A:
  - S=(-37dbm, .., -77dbm), L=(1, 3)
  - S=(-41dbm, .., -83dbm), L=(1, 4)
  - S=(-49dbm, .., -34dbm), L=(9, 10)
  - S=(-61dbm, .., -28dbm), L=(15, 22)
- Device B:
  - S=(-33dbm, .., -82dbm), L=(1, 3)
  - S=(-57dbm, .., -63dbm), L=(10, 23)

Localization model

Test
- Device A:
  - S=(-37dbm, .., -77dbm)
  - S=(-41dbm, .., -83dbm)
  - S=(-49dbm, .., -34dbm)
  - S=(-61dbm, .., -28dbm)

Average Error Distance
- ~1.5 meters
- ~10 meters

Drop!
Difference between Domains

Time Period A

Device A

Time Period B

Device B
Motivating Example II:
Sentiment classification

10 hours ago
Edward Priz replied:
You know, this isn't the first time that "States Rights" has been used as a cover for racist policies. In fact, the whole "States Rights" thing has become a sort of code for heavy-handed racist policies, hasn't it? And it does provide a sort of contextual...

10 hours ago
RICH HIRTH replied:
The issue here is probable cause. A police officer can question if he has probable cause, and he can document it. This law can be abused if being Latino is probable cause. That is license to harass for the police. As long as the law is applied fairly there...

2 hours ago
Julia Gomez replied:
The Arizona law is so clearly unconstitutional that I do not think it will ever reach the point of being enforced. The article did not say so, but the Republican governor is afraid of a GOP primary electorate that is even more reactionary than usual. That is why she signed the bill, not because she thinks it is legally defensible.
Sentiment Classification (cont.)

Training

Sentiment Classifier

Test

Electronics

Electronics

Classification Accuracy

~ 84.6%

Drop!

~ 72.65%
### Difference between Domains

<table>
<thead>
<tr>
<th>Electronics</th>
<th>Video Games</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) <strong>Compact</strong>; easy to operate; very good picture quality; looks <strong>sharp</strong>!</td>
<td>(2) A very good game! It is action packed and full of excitement. I am very much <strong>hooked</strong> on this game.</td>
</tr>
<tr>
<td>(3) I purchased this unit from Circuit City and I was very excited about the quality of the picture. It is really nice and <strong>sharp</strong>.</td>
<td>(4) Very <strong>realistic</strong> shooting action and good plots. We played this and were <strong>hooked</strong>.</td>
</tr>
<tr>
<td>(5) It is also quite <strong>blurry</strong> in very dark settings. I will never buy HP again.</td>
<td>(6) The game is so <strong>boring</strong>. I am extremely unhappy and will probably never buy UbiSoft again.</td>
</tr>
</tbody>
</table>
A Major Assumption in Traditional Machine Learning

- Training and future (test) data come from the same domain, which implies
  - Represented in the same feature spaces.
  - Follow the same data distribution.
In Real-world Applications

• Training and testing data may come from different domains, which have:
  • Different marginal distributions, or different feature spaces:
    \[ \mathcal{X}_S \neq \mathcal{X}_T, \text{ or } P_S(x) \neq P_T(x) \]
  • Different predictive distributions, or different label spaces:
    \[ \mathcal{Y}_S \neq \mathcal{Y}_T, \text{ or } f_S \neq f_T \left( P_S(y|x) \neq P_T(y|x) \right) \]
How to Build Systems on Each Domain of Interest

- Build every system from scratch?
  - Time consuming and expensive!

- Reuse common knowledge extracted from existing systems?
  - More practical!
The Goal of Transfer Learning

Labeled Training

Source Domain Data

Electronics

Time Period A

Device A

Transfer Learning Algorithms

Target Domain Data

Unlabeled Training

Time Period B

Device B

Predictive Models

Target Domain Data

Testing

DVD
Transfer Learning Settings

Transfer Learning

Feature Space

Heterogeneous Transfer Learning

Homogeneous Transfer Learning

Heterogeneous Transfer Learning

Supervised Transfer Learning

Semi-Supervised Transfer Learning

Unsupervised Transfer Learning

Homogeneous Transfer Learning

Heterogeneous Feature Space

Homogeneous Feature Space
Transfer Learning Approaches

- Instance-based Approaches
- Feature-based Approaches
- Parameter-based Approaches
- Relational Approaches
Instance-based Transfer Learning Approaches

General Assumption
Source and target domains have a lot of overlapping features (domains share the same/similar support)
## Instance-based Transfer Learning Approaches

### Case I

**Problem Setting**

Given $D_S = \{x_{S_i}, y_{S_i}\}_{i=1}^{n_S}$, $D_T = \{x_{T_i}\}_{i=1}^{n_T}$,

Learn $f_T$, s.t. $\sum_i \epsilon(f_T(x_{T_i}), y_{T_i})$ is small,

where $y_{T_i}$ is unknown.

**Assumption**

- $\mathcal{Y}_S = \mathcal{Y}_T$, and $P(Y_S|X_S) = P(Y_T|X_T)$,
- $\mathcal{X}_S \approx \mathcal{X}_T$,
- $P(X_S) \neq P(X_T)$.

### Case II

**Problem Setting**

Given $D_S = \{x_{S_i}, y_{S_i}\}_{i=1}^{n_S}$, $D_T = \{x_{T_i}, y_{T_i}\}_{i=1}^{n_T}$, $n_T \ll n_S$,

Learn $f_T$, s.t. $\epsilon(f_T(x_{T_i}), y_{T_i})$ is small, and $f_T$ has good generalization on unseen $x_T^*$.

**Assumption**

- $\mathcal{Y}_S = \mathcal{Y}_T$,
- $\mathcal{X}_S \approx \mathcal{X}_T$,
- $P(X_S) \neq P(X_T)$.
Instance-based Approaches

Case I

Given a target task,

\[
\theta^* = \arg \min E_{(x,y) \sim P_T} [l(x, y, \theta)] \\
= \arg \min E_{(x,y) \sim P_T} \left[ \frac{P_S(x,y)}{P_S(x,y)} l(x, y, \theta) \right] \\
= \arg \min \int_y \int_x P_T(x,y) \left( \frac{P_S(x,y)}{P_S(x,y)} l(x, y, \theta) \right) dx dy \\
= \arg \min \int_y \int_x P_S(x,y) \left( \frac{P_T(x,y)}{P_S(x,y)} l(x, y, \theta) \right) dx dy \\
= \arg \min E_{(x,y) \sim P_S} \left[ \frac{P_T(x,y)}{P_S(x,y)} l(x, y, \theta) \right]
\]
Instance-based Approaches

Case I (cont.)

If $P_S(x, y) = P_T(x, y)$

$$\theta^* = \arg \min \mathbb{E}_{(x_T, y_T) \sim P_T}[l(x_T, y_T, \theta)]$$

$$\theta^* = \arg \min \mathbb{E}_{(x_S, y_S) \sim P_S}[l(x_S, y_S, \theta)]$$

$$\theta^* = \arg \min \sum_{i=1}^{n_S} l(x_{S_i}, y_{S_i}, \theta) + \lambda \Omega(\theta)$$
Instance-based Approaches

Case I (cont.)

Assumption: \( \{ P_S(x) \neq P_T(x), \ P_S(y|x) = P_T(y|x) \} \Rightarrow P_S(x,y) \neq P_T(x,y) \)

\[
\begin{align*}
\theta^* &= \arg \min \mathbb{E}_{(x,y) \sim P_S} \left[ \frac{P_T(x,y)}{P_S(x,y)} l(x,y,\theta) \right] \\
&= \arg \min \mathbb{E}_{(x,y) \sim P_S} \left[ \frac{P_T(x) P_T(y|x)}{P_S(x) P_S(y|x)} l(x,y,\theta) \right] \\
&= \arg \min \mathbb{E}_{(x,y) \sim P_S} \left[ \frac{P_T(x)}{P_S(x)} l(x,y,\theta) \right] \\
\text{Denote } \beta(x) &= \frac{P_T(x)}{P_S(x)}, \\
\theta^* &= \arg \min \sum_{i=1}^{n_S} \beta(x_{S_i}) l(x_{S_i},y_{S_i},\theta) + \lambda \Omega(\theta)
\end{align*}
\]
Instance-based Approaches

Case I (cont.)

How to estimate $\beta(x) = \frac{P_T(x)}{P_S(x)}$?

A simple solution is to first estimate $P_T(x)$, $P_S(x)$, respectively, and calculate $\frac{P_T(x)}{P_S(x)}$. $\times$

An alternative solution is to estimate $\frac{P_T(x)}{P_S(x)}$ directly. $\checkmark$

Correcting Sample Selection Bias / Covariate Shift
[Quionero-Candela, etal, Data Shift in Machine Learning, MIT Press 2009]
Instance-based Approaches

Correcting sample selection bias

- Imagine a rejection sampling process, and view the source domain as samples from the target domain.
Instance-based Approaches
Correcting sample selection bias (cont.)

• The distribution of the selector variable maps the target onto the source distribution

\[ P_S(x) \propto P_T(x)P(s = 1|x) \]

\[ \beta(x) = \frac{P_T(x)}{P_S(x)} \propto \frac{1}{P(s = 1|x)} \]

- Labeled instances from the source domain with label 1
- Unlabeled instances from the target domain with label 0
- Train a binary classifier

[Zadrozny, ICML-04]
Instance-based Approaches

Kernel mean matching (KMM)

Maximum Mean Discrepancy (MMD)

Given $X_S = \{x_{S_i}\}_{i=1}^{n_S}$, $X_T = \{x_{T_i}\}_{i=1}^{n_T}$, drawn from $P_S(x)$ and $P_T(x)$, respectively,

$$\text{Dist}(P(X_S), P(X_T)) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \Phi(x_{S_i}) - \frac{1}{n_T} \sum_{j=1}^{n_T} \Phi(x_{T_j}) \right\|_H$$

[Alex Smola, Arthur Gretton and Kenji Kukumizu, ICML-08 tutorial]
**Instance-based Approaches**

**Kernel mean matching (KMM) (cont.)**

[Huang et al., NIPS-06]

\[
\begin{align*}
\arg\min_{\beta} & \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_{S_i})\Phi(x_{S_i}) - \frac{1}{n_T} \sum_{j=1}^{n_T} \Phi(x_{T_j}) \right\| \\
\text{s.t.} & \quad \beta(x_{S_i}) \in [0, B] \quad \text{and} \quad \left| \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_{S_i}) - 1 \right| \leq \epsilon.
\end{align*}
\]

\[K_{ij} = k(x_{S_i}, x_{S_j})\]

\[
\begin{align*}
\arg\min_{\beta} & \frac{1}{2} \beta^T K \beta - \kappa^T \beta \\
\text{s.t.} & \quad \beta(x_{S_i}) \in [0, B] \quad \text{and} \quad \left| \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_{S_i}) - 1 \right| \leq \epsilon.
\end{align*}
\]

\[\kappa_i = \frac{n_S}{n_T} \sum_{j=1}^{n_T} k(x_{S_i}, x_{T_j})\]
Instance-based Approaches

Direct density ratio estimation

Recall $\beta(x) = \frac{P_T(x)}{P_S(x)}$.

Let $\tilde{\beta}(x) = \sum_{\ell=1}^b \alpha_\ell \psi_\ell(x)$, and denote $\tilde{P}_T(x) = \tilde{\beta}(x)P_S(x)$.

KL divergence loss

$\arg \min_{\{\alpha_\ell\}_{\ell=1}^b} KL[P_T(x) \mid \mid \tilde{P}_T(x)]$

Least squared loss

$\arg \min_{\{\alpha_\ell\}_{\ell=1}^b} \int_{X_S \cup X_T} \left( \frac{\tilde{\beta}(x) - \beta(x)}{\tilde{\beta}(x)P_S(x)} \right)^2 P_S(x)dx$

[Sugiyama et al., NIPS-07]

[Kanamori et al., JMLR-09]
Instance-based Approaches

Case II

- $\mathcal{Y}_S = \mathcal{Y}_T,$
  
  but $f_S \neq f_T \ (P_S(y|x) \neq P_T(y|x)).$

- Intuition: Part of the labeled data in the source domain can be reused in the target domain after re-weighting.
Instance-based Approaches

Case II (cont.)

- **TrAdaBoost** [Dai *etal* ICML-07]
  - For each boosting iteration,
    - Use the same strategy as AdaBoost to update the weights of target domain data.
    - Use a new mechanism to decrease the weights of misclassified source domain data.
Feature-based Transfer Learning Approaches

When source and target domains only have some overlapping features. (lots of features only have support in either the source or the target domain)
Feature-based Transfer Learning Approaches (cont.)

How to learn $\varphi$?

- **Solution 1**: Encode application-specific knowledge to learn the transformation.

- **Solution 2**: General approaches to learning the transformation.
Feature-based Approaches
Encode application-specific knowledge

<table>
<thead>
<tr>
<th>Electronics</th>
<th>Video Games</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) <strong>Compact</strong>; easy to operate; very good picture quality; looks <strong>sharp</strong>!</td>
<td>(2) A very good game! It is action packed and full of excitement. I am very much <strong>hooked</strong> on this game.</td>
</tr>
<tr>
<td>(3) I purchased this unit from Circuit City and I was very excited about the quality of the picture. It is really nice and <strong>sharp</strong>.</td>
<td>(4) Very <strong>realistic</strong> shooting action and good plots. We played this and were <strong>hooked</strong>.</td>
</tr>
<tr>
<td>(5) It is also quite <strong>blurry</strong> in very dark settings. I will never_buy HP again.</td>
<td>(6) The game is so <strong>boring</strong>. I am extremely unhappy and will probably never_buy UbiSoft again.</td>
</tr>
</tbody>
</table>
Feature-based Approaches
Encode application-specific knowledge (cont.)

<table>
<thead>
<tr>
<th>compact</th>
<th>sharp</th>
<th>blurry</th>
<th>hooked</th>
<th>realistic</th>
<th>boring</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Training

\[ y = f(x) = \text{sgn}(w \cdot x^T), \quad w = [1, 1, -1, 0, 0, 0] \]

Prediction

<table>
<thead>
<tr>
<th>compact</th>
<th>sharp</th>
<th>blurry</th>
<th>hooked</th>
<th>realistic</th>
<th>boring</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Feature-based Approaches
Encode application-specific knowledge (cont.)

<table>
<thead>
<tr>
<th>Electronics</th>
<th>Video Games</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) <strong>Compact</strong>; easy to operate; very <strong>good</strong> picture quality; looks <strong>sharp</strong>!</td>
<td>(2) A very <strong>good</strong> game! It is action packed and full of <strong>excitement</strong>. I am very much <strong>hooked</strong> on this game.</td>
</tr>
<tr>
<td>(3) I purchased this unit from Circuit City and I was very <strong>excited</strong> about the quality of the picture. It is really <strong>nice</strong> and <strong>sharp</strong>.</td>
<td>(4) Very <strong>realistic</strong> shooting action and <strong>good</strong> plots. We played this and were <strong>hooked</strong>.</td>
</tr>
<tr>
<td>(5) It is also quite <strong>blurry</strong> in very dark settings. I will <strong>never buy</strong> HP again.</td>
<td>(6) The game is so <strong>boring</strong>. I am extremely <strong>unhappy</strong> and will probably <strong>never buy</strong> UbiSoft again.</td>
</tr>
</tbody>
</table>
Feature-based Approaches
Encode application-specific knowledge (cont.)

- Three different types of features
  - Source domain (*Electronics*) specific features, e.g., *compact, sharp, blurry*
  - Target domain (*Video Game*) specific features, e.g., *hooked, realistic, boring*
  - Domain independent features (pivot features), e.g., *good, excited, nice, neverBuy*
Feature-based Approaches
Encode application-specific knowledge (cont.)

- How to identify *pivot* features?
  - Term frequency on both domains
  - Mutual information between features and labels (source domain)
  - Mutual information on between features and domains

- How to utilize pivots to *align* features across domains?
  - Structural Correspondence Learning (SCL) [Biltzer *etal.* EMNLP-06]
  - Spectral Feature Alignment (SFA) [Pan *etal.* WWW-10]
Feature-based Approaches
Structural Correspondence Learning (SCL)

Intuition

- Use *pivot* features to construct *pseudo* tasks that are related to the target classification task.
- Model correlations between *pivot* features and other features using multi-task learning techniques.
- Discover new shared features by exploiting the feature correlations.
Structural Correspondence Learning
Algorithm

- Identify $P$ pivot features
- Build $P$ classifiers to predict the pivot features from remaining features
- Discover *shared* feature subspace
  - Compute top $K$ *eigenvectors*
  - Project original features into eigenvectors to derive new shared features
- Train classifiers on the source using *augmented* features (original features + new features)
Feature-based Approaches
Spectral Feature Alignment (SFA)

➤ Intuition

- Use a bipartite graph to model the correlations between pivot features and other features
- Discover new shared features by applying spectral clustering techniques on the graph
Spectral Feature Alignment (SFA)

High level idea

- If two *domain-specific* words have connections to more common *pivot* words in the graph, they tend to be aligned or clustered together with a higher probability.
- If two *pivot* words have connections to more common *domain-specific* words in the graph, they tend to be aligned together with a higher probability.
Derive new features

**Pivot features**

- exciting
- good
- never_buy

**Domain-specific features**

- realistic
- compact
- hooked
- sharp
- blurry
- boring

**Spectral Clustering**

- Electronics
- Video Game

**Video Game**
- boring
- blurry

**Electronics**
- compact
- realistic
- sharp
- hooked
Spectral Feature Alignment (SFA)

Derive new features (cont.)

<table>
<thead>
<tr>
<th>sharp/hooked</th>
<th>compact/realistic</th>
<th>blurry/boring</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Electronics

\[ y = f(x) = \text{sgn}(w \cdot x^T), \quad w = [1, 1, -1] \]

Video Game

Training

Prediction
Spectral Feature Alignment (SFA) Algorithm

- Identify $P$ pivot features
- Construct a bipartite graph between the pivot and remaining features.
- Apply spectral clustering on the graph to derive new features
- Train classifiers on the source using augmented features (original features + new features)
Feature-based Approaches

Develop general approaches

Time Period A

Device A

Time Period B

Device B
Feature-based Approaches

General approaches

- Unsupervised feature learning
- Semi-supervised feature learning
- Multi-task feature learning
- Self-taught feature learning
Feature-based Approaches

Transfer Component Analysis [Pan et al., IJCAI-09, TNN-11]

Motivation

Latent factors

Source

Target

Temperature

Signal properties

Power of APs

Building structure
Transfer Component Analysis (cont.)

Latent factors

Source

Target

Temperature
Signal properties
Power of APs
Building structure

Cause the data distributions between domains different
Transfer Component Analysis (cont.)

Source

Noisy component

Signal properties

Principal components

Target

Building structure
Transfer Component Analysis (cont.)

Learning $\varphi$ by only minimizing distance between distributions may map the data onto noisy factors.
Main idea: the learned $\varphi$ should map the source and target domain data to the latent space spanned by the factors which can reduce domain difference and preserve original data structure.

High level optimization problem

$$\min_{\varphi} \text{Dist}(\varphi(X_S), \varphi(X_T)) + \lambda \Omega(\varphi)$$

s.t. constraints on $\varphi(X_S)$ and $\varphi(X_T)$
Recall: Maximum Mean Discrepancy (MMD)

Given $X_S = \{x_{S_i}\}_{i=1}^{n_S}$, $X_T = \{x_{T_i}\}_{i=1}^{n_T}$, drawn from $P_S(x)$ and $P_T(x)$, respectively,

$$\text{Dist}(P(X_S), P(X_T)) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \Phi(x_{S_i}) - \frac{1}{n_T} \sum_{j=1}^{n_T} \Phi(x_{T_j}) \right\|_{\mathcal{H}}$$
Transfer Component Analysis (cont.)

\[
\text{Dist}(\varphi(X_S), \varphi(X_T)) = \left\| \mathbb{E}_{x \sim P_T(x)}[\Phi(\varphi(x))] - \mathbb{E}_{x \sim P_S(x)}[\Phi(\varphi(x))] \right\|
\]

\[
\approx \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \Phi(\varphi(x_{S_i})) - \frac{1}{n_T} \sum_{i=1}^{n_T} \Phi(\varphi(x_{T_i})) \right\|
\]

Assume \( \Psi = \Phi \circ \varphi \) a RKHS, with kernel \( k(x_i, x_j) = \Psi(x_i)^	op \Psi(x_j) \)

\[
\text{Dist}(\varphi(X_S), \varphi(X_T)) = \text{tr}(KL)
\]

\[
K = \begin{bmatrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{bmatrix} \in \mathbb{R}^{(n_S+n_T) \times (n_S+n_T)},
L_{ij} = \begin{cases} 
\frac{1}{n_S}, & x_i, x_j \in X_S, \\
\frac{1}{n_T}, & x_i, x_j \in X_T, \\
-\frac{1}{n_S n_T}, & \text{otherwise}.
\end{cases}
\]
Transfer Component Analysis (cont.)

\[
\begin{align*}
\min_{\varphi} & \quad \text{Dist}(\varphi(X_S), \varphi(X_T)) + \lambda \Omega(\varphi) \\
\text{s.t.} & \quad \text{constraints on } \varphi(X_S) \text{ and } \varphi(X_T) \\
\min_{\varphi} & \quad \text{tr}(KL) + \lambda \Omega(\varphi) \\
\text{s.t.} & \quad \text{constraints on } \varphi(X_S) \text{ and } \varphi(X_T)
\end{align*}
\]

- The kernel function can be a highly nonlinear function of $\varphi$
- A direct optimization of minimizing the quantity w.r.t. $\varphi$ can get stuck in poor local minima
Transfer Component Analysis (cont.)

To maximize the data variance
To minimize the distance between domains
To preserve the local geometric structure

\[ \text{Learning } \varphi \Rightarrow (1) \text{ learning } K \]

\[ \text{(2) low-dimensional reconstructions of } X_S \text{ and } X_T \]

\[ \text{based on } K \]

\[ \text{Learning } K \Rightarrow \min_{K \succeq 0} \text{tr}(KL) - \lambda \text{tr}(K) \]

\[ \text{s.t. } K_{ii} + K_{jj} - 2K_{ij} = d_{ij}^2, \forall (i, j) \in N. \]

\[ K1 = 0, K \succeq 0. \]

Low-dimensional constructions of \( X_S, X_T \Rightarrow \) PCA on \( K \)

- It is a SDP problem, expensive!
- It is transductive, cannot generalize on unseen instances!
- PCA is post-processed on the learned kernel matrix, which may potentially discard useful information.
Transfer Component Analysis (cont.)

Decompose $K = (KK^{-1/2})(K^{-1/2}K)$

Let $\tilde{W} \in \mathbb{R}^{(n_S+n_T) \times m}$, where $m \ll n_S + n_T$.

$\tilde{K} = (KK^{-1/2}\tilde{W})(\tilde{W}^\top K^{-1/2}K) = KWW^\top K$,

$W = K^{-1/2}\tilde{W} \in \mathbb{R}^{(n_S+n_T) \times m}$.

\forall x_i, x_j, \tilde{k}(x_i, x_j) = k_{x_i}^\top WW^\top k_{x_j},$

where $k_x = [k(x_1, x), \ldots, k(x_{n_S+n_T}, x)]^\top \in \mathbb{R}^{n_S+n_T}$.

Learning $\varphi \Rightarrow$ learning a low-rank matrix $W$
Transfer Component Analysis (cont.)

To minimize the distance between domains

$$\min_W \text{tr}(W^T KKLKW) + \lambda \text{tr}(W^T W)$$

subject to

$$W^T KHKW = I.$$ 

Regularization on W

To maximize the data variance

$$W^* \iff m \text{ leading eigenvectors of } (KLK + \lambda I)^{-1}KHK,$$

where $m \leq n_S + n_T - 1.$
Transfer Component Analysis (cont.)

An illustrative example
Latent features learned by PCA and TCA

Original feature space

PCA

TCA
Transfer Component Analysis

Limitations

Unsupervised TCA

Con: the direction with the largest data variance may be different or even orthogonal (the worst case) to the discriminative direction.

Solution: to borrow an idea from kernel target alignment [Cristianini et al. NIPS-02] to maximize the dependence between features and labels
Semi-Supervised Transfer Component Analysis

[Pan et al., TNN-11]

High level objectives:

\[
\begin{align*}
\min_\varphi & \quad \text{Dist}(\varphi(X_S), \varphi(X_T)) + \lambda \Omega(\varphi), \\
\max_\varphi & \quad \text{Dep}(\varphi(X_S), Y_S), \\
\text{s.t.} & \quad \text{Constraints on } \varphi(X_S) \text{ and } \varphi(X_T)
\end{align*}
\]

To measure label dependence using Hilbert-Schmidt Independence Criterion (HSIC)

To measure the distance between domains using MMD
Hilbert-Schmidt Independence Criterion (HSIC)

[Alex Smola, Arthur Gretton and Kenji Kukumizu, ICML-08 tutorial]

\[
\text{HSIC}(X_S, Y_S) = \frac{1}{n_S^2} \left\| \sum_{i,j=1}^{n_S,n_S} \left[ \left( \psi_1(x_{S_i}) - \frac{1}{n_S} \sum_{k=1}^{n_S} \psi_1(x_{S_k}) \right) \otimes \left( \psi_2(y_{S_j}) - \frac{1}{n_S} \sum_{k=1}^{n_S} \psi_2(y_{S_k}) \right) \right] \right\|_\mathcal{H}
\]

Trace form:

\[
\text{Dep}(\varphi(X_S), Y_S) = \text{HSIC}(\varphi(X_S), Y_S) = \frac{1}{(n_S - 1)^2} \text{tr}(HKHK_{yy}),
\]

where \( K, K_{yy} \) are kernel matrices defined on \( \varphi(X_S) \) and \( Y_S \),

\[
H = I - \frac{1}{n_S} 11^T
\]

is the centering matrix.
Semi-Supervised Transfer Component Analysis (cont.)

Algorithm: generalized eigen decomposition

\[
\begin{align*}
\min_W & \quad \text{tr}(W^T K L K W) + \lambda_1 \text{tr}(W^T W) + \frac{\lambda_2}{(n_S + n_T)^2} \text{tr}(W^T K L K W) \\
\text{s.t.} & \quad W^T K H K_{yy} H K W = I
\end{align*}
\]

- To minimize the distance between domains
- To maximize the data variance
- To maximize the label dependence
- To control the complexity of W
- To preserve geometric locality
- To propagate label information

Laplacian
Feature-based Approaches
Multi-task Feature Learning

General Multi-task Learning Setting

Given $D_S = \{x_{Si}, y_{Si}\}_{i=1}^{n_S}$, $D_T = \{x_{Ti}, y_{Ti}\}_{i=1}^{n_T}$, where $n_S$ and $n_T$ are small,

Learn $f_S, f_T$, s.t. $\sum_{t \in \{S,T\}} \sum_i \epsilon(f_t(x_{ti}), y_{ti})$ is small.

- **Assumption:** If tasks are related, they should share some **good** common features.
- **Goal:** Learn a low-dimensional representation shared across related tasks.
Feature-based Approaches
Multi-task Feature Learning (cont.)

Assume \( f(x) = \langle \theta, (U^T x) \rangle = \theta^T (U^T x) \), where \( \theta \in \mathbb{R}^k, x \in \mathbb{R}^m, U \in \mathbb{R}^{m \times k} \)

\[
\{\Theta^*, U^*\} = \arg \min_{t \in \{S,T\}} \sum_{i=1}^{n_t} l(U^T x_{t_i}, y_{t_i}, \theta_t) + \lambda_1 \Omega(\Theta)
\]

s.t. constraints on \( U \).

\( \Theta = [\theta_S, \theta_T] \in \mathbb{R}^{k \times 2} \)
Feature-based Approaches
Multi-task Feature Learning (cont.)

- Solution 1: $U$ is full rank ($U \in \mathbb{R}^{m \times k}$, $k = m$), $\Theta$ is sparse.

$$\arg\min_{\Theta, U} \sum_{t \in \{S, T\}} \sum_{i=1}^{n_t} l(U^T x_{t_i}, y_{t_i}, \theta_t) + \lambda \|\Theta\|_{2,1}$$

s.t. $U \in \mathbb{O}^m$

$$U^T U = UU^T = I$$

$$\|\Theta\|_{2,1} = \sum_{i}^{m} \|\Theta^i\|_2^1$$, where $\Theta^i$ is the $i^{th}$ row of $\Theta$.

[Argyriou etal., NIPS-07]
Feature-based Approaches
Multi-task Feature Learning (cont.)

Solution 1: $U$ is full rank ($U \in \mathbb{R}^{m \times k}, k = m$), $\Theta$ is sparse.

$$\arg \min_{\Theta, U} \sum_{t \in \{S,T\}} \sum_{i=1}^{n_t} l(U^T x_{t_i}, y_{t_i}, \theta_t) + \lambda \|\Theta\|_{2,1}$$

s.t. $U \in \mathcal{O}^m$.

Illustration
Feature-based Approaches
Multi-task Feature Learning (cont.)

- Solution 2: $U$ is low rank ($U \in \mathbb{R}^{m \times k}$, $k < m$, in practice $k \ll m$).

$$\arg\min_{\Theta, U} \sum_{t \in \{S, T\}} \sum_{i=1}^{n_t} l(\tilde{x}_{t_i}, y_{t_i}, \theta_t) + \lambda \Omega(\Theta, U)$$

s.t. $$U^T U = I_{k \times k},$$

where $$\tilde{x}_{t_i}^T = [x_{t_i}^T, (U^T x_{t_i})^T], \quad \theta_t^T = [w_t^T, v_t^T].$$
Feature-based Approaches
Multi-task Feature Learning (cont.)

Solution 2: \( U \) is low rank (\( U \in \mathbb{R}^{m \times k}, k < m, \) in practice \( k \ll m \)).

\[
\arg \min_{\Theta, U} \sum_{t \in \{S, T\}} \sum_{i=1}^{n_t} l(\tilde{x}_{t_i}, y_{t_i}, \theta_t) + \lambda \Omega(\Theta, U)
\]

s.t. \( U^T U = I_{k \times k} \),

where \( \tilde{x}_{t_i}^T = [x_{t_i}^T, (U^T x_{t_i})^T] \), \( \theta_t^T = [w_t^T, v_t^T] \).

[Ando and Zhang, JMLR-05] \( \Omega(\Theta, U) = \sum_{t \in \{S, T\}} ||w_t||_2^2 \)

[Ji et al., KDD-08] \( \Omega(\Theta, U) = \sum_{t \in \{S, T\}} (\lambda_1 ||w_t||_2^2 + \lambda_2 ||w_t + U v_t||_2^2) \)
Feature-based Approaches
Self-taught Feature Learning

- **Intuition:** There exist some higher-level features that can help the target learning task even only a few labeled data are given.

- **Steps:**
  1) Learn higher-level features from a lot of unlabeled data.
  2) Use the learned higher-level features to represent the data of the target task.
  3) Training models from the new representations of the target task with corresponding labels.
Feature-based Approaches
Self-taught Feature Learning

How to learn higher-level features
- Sparse Coding [Raina et al., 2007]
- Deep learning [Glorot et al., 2011]
Parameter-based Transfer Learning Approaches

Motivation: A well-trained model $\theta_S^*$ has learned a lot of structure. If two tasks are related, this structure can be transferred to learn $\theta_T^*$. 

Assume $f(x) = \langle \theta, x \rangle = \theta^T x = \sum_{i=1}^{m} \theta_i x_i$, where $\theta, x \in \mathbb{R}^m$.

$$\theta_S^* = \arg \min_{\theta_S} \sum_{i=1}^{n_S} l(x_{S_i}, y_{S_i}, \theta_S) + \lambda \Omega(\theta_S).$$

$$\theta_T^* = \arg \min_{\theta_T} \sum_{i=1}^{n_T} l(x_{T_i}, y_{T_i}, \theta_T) + \lambda \Omega(\theta_T).$$

Tasks are learned independently
Parameter-based Approaches
Multi-task Parameter Learning

Assumption:
If tasks are related, they may share similar parameter vectors. For example, [Evgeniou and Pontil, KDD-04]

\[
\begin{align*}
\theta_S &= \theta_0 + \nu_S, \\
\theta_T &= \theta_0 + \nu_T,
\end{align*}
\]

\[
\{\theta_S^*, \theta_T^*\} = \arg\min_{\theta_S, \theta_T} \sum_{t \in \{S,T\}} \sum_{i=1}^{n_t} l(x_{ti}, y_{ti}, \theta_t) + \lambda \Omega(\theta_0, \nu_S, \nu_T)
\]
Parameter-based Approaches

Multi-task Parameter Learning (cont.)

\[
\{\theta_S^*, \theta_T^*\} = \arg \min \sum_{t \in \{S,T\}} \sum_{i=1}^{n_t} l(x_{ti}, y_{ti}, \theta_t) + \lambda \Omega(\theta_0, v_S, v_T)
\]

\[
= \arg \min \sum_{t \in \{S,T\}} \sum_{i=1}^{n_t} l(x_{ti}, y_{ti}, \theta_t) + \frac{\lambda_1}{2} \sum_{t \in \{S,T\}} \Omega(v_t) + \lambda_2 \Omega(\theta_0)
\]

\[
= \arg \min \sum_{t \in \{S,T\}} \sum_{i=1}^{n_t} l(x_{ti}, y_{ti}, \theta_t) + \frac{\lambda_1}{2} \sum_{t \in \{S,T\}} \|v_t\|^2 + \lambda_2 \|\theta_0\|^2
\]
Parameter-based Approaches
Multi-task Parameter Learning (cont.)

A general framework:

Denote $\Theta = [\theta_S, \theta_T]$,

$$\Theta^* = \arg\min \sum_{t \in \{S,T\}} \sum_{i=1}^{n_t} l(x_{t_i}, y_{t_i}, \theta_t) + \lambda_1 \text{tr}(\Theta^T \Theta) + \lambda_2 f(\Theta)$$

$$f(\Theta) = \sum_{t \in \{S,T\}} \left\| \theta_t - \frac{1}{2} \sum_{s \in \{S,T\}} \theta_s \right\|^2$$

$$\sum_{t \in \{S,T\}} \left\| \theta_t \right\|^2$$

[Zhang and Yeung, UAI-10]

$$f(\Theta) = \text{tr}(\Theta^T \Sigma^{-1} \Theta)$$

s.t. $\Sigma \succeq 0$ and $\text{tr}(\Sigma) = 1$.

[Agarwal et al, NIPS-10]

$$f(\Theta) = \sum_{t \in \{S,T\}} \left\| \theta_t - \tilde{\theta}_t^M \right\|^2$$
Relational Transfer Learning Approaches

Motivation: If two relational domains (data is non-i.i.d) are related, they may share some similar relations among objects. These relations can be used for knowledge transfer across domains.
Relational Transfer Learning Approaches (cont.)

Academic domain (source)

Student (B) \xrightarrow{\text{AdvisedBy}} \text{Professor (A)}

\text{Publication} \quad \text{Publication}

\text{Paper (T)}

Movie domain (target)

Actor (A) \xrightarrow{\text{WorkedFor}} \text{Director (B)}

\text{MovieMember} \quad \text{MovieMember}

\text{Movie (M)}

\text{AdvisedBy (B, A) \land \text{Publication (B, T)} \Rightarrow \text{Publication (A, T)}}

\text{WorkedFor (A, B) \land \text{MovieMember (A, M)} \Rightarrow \text{MovieMember (B, M)}}

P_1(x, y) \land P_2(x, z) \Rightarrow P_2(y, z)
Relational Transfer Learning Approaches

- Predicate mapping and revising
  [Mihalkova et al., AAAI-07]

- Second-order Markov Logic
  [Davis and Domingos, ICML-09]
The **camera** is **great**.
It is a very **amazing** **product**.
I highly **recommend** this **camera**.
It takes excellent **photos**.
**Photos** had some **artifacts** and **noise**.

**Task**: sentiment summarization

- What is the opinion expressed on?
  - To construct lexicon of **topic** or **target** words
- How is the opinion expressed?
  - To construct lexicon of **sentiment** words
Relational Approaches
Relational Adaptive bootstrapping (RAP) (cont.)

The **camera** is great.
It is a very **amazing** product.
I highly recommend this **camera**.
It takes **excellent photos**.
**Photos** had some **artifacts** and **noise**.

This **movie** has **good script**, **great casting**, **excellent acting**.
This **movie** is so **boring**.
The **Godfather** was the most **amazing movie**.
The **movie** is **excellent**.
Relational Approaches

RAP (cont.)

➤ Bridge between cross-domain sentiment words
  – Domain independent (general) sentiment words

➤ Bridge between cross-domain topic words
Relational Approaches

RAP (cont.)

Bridge between cross-domain topic words

- Syntactic structure between topic and sentiment words

Common syntactic pattern: “topic word” – nsubj – “sentiment word”
Summary

Transfer Learning

Heterogeneous Transfer Learning

Supervised Transfer Learning

Semi-Supervised Transfer Learning

Unsupervised Transfer Learning

Homogeneous Transfer Learning

Instance-based Approaches

Feature-based Approaches

Relational Approaches

Parameter-based Approaches

In data level

In model level
Some Research Issues in Transfer Learning

- How to avoid negative transfer? Given multiple source domains, how to find good ones to ensure positive transfer in the target domain.

- How to optimize performance measures for transfer learning

- How to transfer knowledge across heterogeneous feature spaces
Reference

- [Thorndike and Woodworth, The Influence of Improvement in one mental function upon the efficiency of the other functions, 1901]
- [Taylor and Stone, Transfer Learning for Reinforcement Learning Domains: A Survey, JMLR 2009]
- [Pan and Yang, A Survey on Transfer Learning, IEEE TKDE 2010]
- [Quionero-Candela, etal, Data Shift in Machine Learning, MIT Press 2009]
- [Biltzer etal., Domain Adaptation with Structural Correspondence Learning, EMNLP 2006]
- [Pan etal., Cross-Domain Sentiment Classification via Spectral Feature Alignment, WWW 2010]
- [Pan etal., Transfer Learning via Dimensionality Reduction, AAAI 2008]
Reference (cont.)

- [Pan et al., Domain Adaptation via Transfer Component Analysis, IJCAI 2009]
- [Evgeniou and Pontil, Regularized Multi-Task Learning, KDD 2004]
- [Zhang and Yeung, A Convex Formulation for Learning Task Relationships in Multi-Task Learning, UAI 2010]
- [Agarwal et al., Learning Multiple Tasks using Manifold Regularization, NIPS 2010]
- [Argyriou et al., Multi-Task Feature Learning, NIPS 2007]
- [Ando and Zhang, A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data, JMLR 2005]
- [Ji et al., Extracting Shared Subspace for Multi-label Classification, KDD 2008]
Reference (cont.)

- [Raina et al., Self-taught Learning: Transfer Learning from Unlabeled Data, ICML 2007]
- [Dai et al., Boosting for Transfer Learning, ICML 2007]
- [Glorot et al., Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach, ICML 2011]
- [Davis and Domingos, Deep Transfer vis Second-order Markov Logic, ICML 2009]
- [Mihalkova et al., Mapping and Revising Markov Logic Networks for Transfer Learning, AAAI 2007]
- [Li et al., Cross-Domain Co-Extraction of Sentiment and Topic Lexicons, ACL 2012]
Reference (cont.)

- [Sugiyama et al., Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation, NIPS 2007]
- [Kanamori et al., A Least-squares Approach to Direct Importance Estimation, JMLR 2009]
- [Cristianini et al., On Kernel Target Alignment, NIPS 2002]
- [Huang et al., Correcting Sample Selection Bias by Unlabeled Data, NIPS 2006]
- [Zadrozny, Learning and Evaluating Classifiers under Sample Selection Bias, ICML 2004]
Thank You