Transfer Learning with Applications

Sinno Jialin Pan\textsuperscript{1}, Qiang Yang\textsuperscript{2,3} and Wei Fan\textsuperscript{3}

\textsuperscript{1} Institute for Infocomm Research, Singapore
\textsuperscript{2} Hong Kong University of Science and Technology
\textsuperscript{3} Huawei Noah's Ark Research Lab, Hong Kong
Outline

- **Part I:** An overview of transfer learning – (Sinno J. Pan)
- **Part II:** Transfer learning applications (Prof. Qiang Yang)
- **Part III:** Advanced research topics: heterogeneous transfer learning (Wei Fan)
Transfer Learning
Overview

Sinno Jialin Pan (Ph.D.)
Lab Head, Text Analytics,
Data Analytics Department,
Institute for Infocomm Research (I2R), Singapore
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

<table>
<thead>
<tr>
<th>Traditional Machine Learning</th>
<th>Transfer Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>training domains</td>
<td>training items</td>
</tr>
<tr>
<td>test domains</td>
<td>test items</td>
</tr>
</tbody>
</table>

domain A  domain B  domain C
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]

Focus!
Motivating Example I:
Indoor WiFi localization

-30dBm -70dBm -40dBm
Indoor WiFi Localization (cont.)

**Training**
- Device A: $S=(-37\text{dbm}, .., -77\text{dbm}), L=(1, 3)$
- $S=(-41\text{dbm}, .., -83\text{dbm}), L=(1, 4)$
- $S=(-49\text{dbm}, .., -34\text{dbm}), L=(9, 10)$
- $S=(-61\text{dbm}, .., -28\text{dbm}), L=(15, 22)$

**Test**
- Device A: $S=(-37\text{dbm}, .., -77\text{dbm})$
- $S=(-41\text{dbm}, .., -83\text{dbm})$
- $S=(-49\text{dbm}, .., -34\text{dbm})$
- $S=(-61\text{dbm}, .., -28\text{dbm})$

**Localization model**

**Average Error Distance**
- ~1.5 meters
- ~10 meters

Drop!
Difference between Domains

Time Period A

Device A

Device B

Time Period 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

Electronics

Sentiment Classifier

Test

Electronics

Drop!

Classification Accuracy

~ 84.6%

~72.65%

Training

DVD

Electronics

Sentiment Classifier

Test

Electronics

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:
    \[ \mathcal{Y}_S \neq \mathcal{Y}_T, \text{ or } f_S \neq f_T \quad (P_S(y|x) \neq P_T(y|x)) \]
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

Transfer Learning Algorithms

Target Domain Data

Labeled Training

Source Domain Data

Predictive Models

Unlabeled data/a few labeled data for adaptation

Time Period A

Electronics

Device A

Time Period B

Target Domain Data

Testing

Device B

DVD
Transfer Learning Settings

- Heterogeneous Transfer Learning
- Supervised Transfer Learning
- Semi-Supervised Transfer Learning
- Unsupervised Transfer Learning

Transfer Learning

Feature Space

Homogeneous

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

Case I

Given a target task,

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

\[
= \arg \min \mathbb{E}_{(x,y) \sim P_T} \left[ \frac{P_S(x,y)}{P_S(x,y)} l(x, y, \theta) \right]
\]

\[
= \arg \min \int_x \int_y P_T(x,y) \left( \frac{P_S(x,y)}{P_S(x,y)} l(x, y, \theta) \right) dx dy
\]

\[
= \arg \min \int_x \int_P P_S(x,y) \left( \frac{P_T(x,y)}{P_S(x,y)} l(x, y, \theta) \right) dx dy
\]

\[
= \arg \min \mathbb{E}_{(x,y) \sim P_S} \left[ \frac{P_T(x,y)}{P_S(x,y)} l(x, y, \theta) \right]
\]
### Instance-based Approaches

#### Case I (cont.)

<table>
<thead>
<tr>
<th>If $P_S(x, y) = P_T(x, y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta^* = \arg \min \mathbb{E}_{(x_T, y_T) \sim P_T}[l(x_T, y_T, \theta)]$</td>
</tr>
<tr>
<td>$\theta^* = \arg \min \mathbb{E}_{(x_S, y_S) \sim P_S}[l(x_S, y_S, \theta)]$</td>
</tr>
<tr>
<td>$\theta^* = \arg \min \sum_{i=1}^{n_S} l(x_{S_i}, y_{S_i}, \theta) + \lambda \Omega(\theta)$</td>
</tr>
</tbody>
</table>
Instance-based Approaches

Case I (cont.)

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

\[
\theta^* = \arg \min_{\theta} \mathbb{E}_{(x,y) \sim P_S} \left[ \frac{P_T(x,y)}{P_S(x,y)} l(x,y,\theta) \right]
\]

\[
= \arg \min_{\theta} \mathbb{E}_{(x,y) \sim P_S} \left[ \frac{P_T(x) P_T(y|x)}{P_S(x) P_S(y|x)} l(x,y,\theta) \right]
\]

\[
= \arg \min_{\theta} \mathbb{E}_{(x,y) \sim P_S} \left[ \frac{P_T(x)}{P_S(x)} l(x,y,\theta) \right]
\]

Denote \( \beta(x) = \frac{P_T(x)}{P_S(x)} \),

\[
\theta^* = \arg \min_{\theta} \sum_{i=1}^{n_S} \beta(x_{S_i}) l(x_{S_i}, y_{S_i}, \theta) + \lambda \Omega(\theta)
\]
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 $\frac{P_T(x)}{P_S(x)}$. ❌

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

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.

Assumption: sample selection bias is caused by the data generation process.
Instance-based Approaches
Correcting sample selection bias (cont.)

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

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

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

- Label instances from the source domain with label 1
- Label 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,

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

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

Kernel mean matching (KMM) (cont.)

[Huang et al., NIPS-06]

$$\arg\min_{\beta} \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_{S_i}) \Phi(x_{S_i}) - \frac{1}{n_T} \sum_{j=1}^{n_T} \Phi(x_{T_j}) \right\|$$

s.t. \( \beta(x_{S_i}) \in [0, B] \) and \( \left| \frac{1}{n_S} \sum_{i=1}^{n_S} \beta(x_{S_i}) - 1 \right| \leq \epsilon. \)
Instance-based Approaches

Direct density ratio estimation

[Sugiyama et al., NIPS-07, Kanamori et al., JMLR-09]

\[ \beta(x) = \frac{P_T(x)}{P_S(x)} \]

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

KL divergence loss

\[
\arg\min_{\{\alpha_{\ell}\}_{\ell=1}^{b}} \text{KL}[P_T(x) || \widetilde{P}_T(x)]
\]

[Sugiyama et al., NIPS-07]

Least squared loss

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

[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 et al. 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_purchase 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>

Electronics

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, never_buy
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 related to 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

Video Game

Electronics

Electronics

Video Game

Electronics

Video Game

Electronics

Video Game

Electronics

Video Game
## Spectral Feature Alignment (SFA)

Derive new features (cont.)

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

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

### 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

Time Period B

Device A

Device B
Feature-based Approaches

General approaches

- Learning features by minimizing distance between distributions
- Learning features inspired by multi-task learning
- Learning features inspired by self-taught 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.)

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

- Source
- Target
- Noisy component
- Signal properties
- Building structure
- Principal components
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 $X_S = \{x_{S_i}\}^{n_S}_{i=1}$, $X_T = \{x_{T_i}\}^{n_T}_{i=1}$, drawn from $P_S(x)$ and $P_T(x)$, respectively,

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

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

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

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

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

- 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.)

Learning $\varphi \Rightarrow (1)$ learning $K$

(2) low-dimensional reconstructions of $X_S$ and $X_T$

based on $K$

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

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

$K1 = 0, K \succeq 0.$

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.

[Pan et al., AAAI-08]
Transfer Component Analysis (cont.)

\[ K = \tilde{K}WW^T\tilde{K}, \text{ where } W \in \mathbb{R}^{(n_S+n_T)\times m} \text{ and } m \ll n_S + n_T. \]

Learning \( K \) \( \Rightarrow \) learning a low-rank matrix \( W \)

\[
\min_W \quad \text{tr}(W^T\tilde{K}L\tilde{K}W) + \lambda \text{tr}(W^TW)
\]

s.t. \( W^T\tilde{K}H\tilde{K}W = I \)

\[ W^* \Leftrightarrow m \text{ leading eigenvectors of } (\tilde{K}L\tilde{K} + \lambda I)^{-1}\tilde{K}H\tilde{K} \]

- Parametric kernel
- Regularization term
- Minimize distance between domains
- Maximize data variance
Transfer Component Analysis (cont.)

An illustrative example
Latent features learned by PCA and TCA

Original feature space

PCA

TCA
Feature-based Approaches
Multi-task Feature Learning

**General Multi-task Learning Setting**

Given \( D_S = \{x_{S_i}, y_{S_i}\}_{i=1}^{n_S} \), \( D_T = \{x_{T_i}, y_{T_i}\}_{i=1}^{n_T} \),
where \( n_S \) and \( n_T \) are small,

Learn \( f_S, f_T \), s.t. \( \sum_{t\in\{S,T\}} \sum_{i} \epsilon(f_t(x_{t_i}), y_{t_i}) \) 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 \( f(x) = \langle \theta, (U^\top x) \rangle = \theta^\top (U^\top x) \), where \( \theta \in \mathbb{R}^k \), \( x \in \mathbb{R}^m \), \( U \in \mathbb{R}^{m \times k} \)

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

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

\( \Theta = [\theta_S, \theta_T] \in \mathbb{R}^{k \times 2} \)

\( U \) is full rank \( (U \in \mathbb{R}^{m \times k}, k = m) \), \( \Theta \) is sparse.  [Argyriou etal., NIPS-07]

\( U \) is low rank \( (U \in \mathbb{R}^{m \times k}, k \ll m) \).  [Ando and Zhang, JMLR-05]

[Ji etal, KDD-08]
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]
Tasks are learned independently

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

\[
\theta^*_S = \arg \min \sum_{i=1}^{n_S} l(x_{S_i}, y_{S_i}, \theta_S) + \lambda \Omega(\theta_S).
\]

\[
\theta^*_T = \arg \min \sum_{i=1}^{n_T} l(x_{T_i}, y_{T_i}, \theta_T) + \lambda \Omega(\theta_T).
\]
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]

\[
\begin{align*}
\theta_S &= \theta_0 + \nu_S \\
\theta_T &= \theta_0 + \nu_T \\
\{\theta^*_S, \theta^*_T\} &= \arg \min \sum_{t \in \{S,T\}} \sum_{i=1}^{n_t} l(x_{ti}, y_{ti}, \theta_t) + \lambda \Omega(\theta_0, \nu_S, \nu_T)
\end{align*}
\]
Parameter-based Approaches
Multi-task Parameter Learning (cont.)

A general framework:

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

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

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

[Zhang and Yeung, UAI-10]

$$
f(\Theta) = \text{tr}(\Theta^T \Sigma^{-1} \Theta)
\text{ s.t. } \Sigma \succeq 0 \text{ and } \text{tr}(\Sigma) = 1.
$$

[Agarwal et al, NIPS-10]

$$
f(\Theta) = \sum_{t \in \{S,T\}} \|\theta_t - \tilde{\theta}_t^M\|^2
$$
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.)

[Mihalkova et al., AAAI-07, Davis and Domingos, ICML-09]

Academic domain (source)

- Student (B) AdvisedBy Professor (A)
- Publication
- Paper (T)

Movie domain (target)

- Actor (A) WorkedFor Director (B)
- MovieMember
- Movie (M)

AdvisedBy (B, A) $\land$ Publication (B, T) $\Rightarrow$ Publication (A, T)

WorkedFor (A, B) $\land$ MovieMember (A, M) $\Rightarrow$ MovieMember (B, M)

P1(x, y) $\land$ P2(x, z) $\Rightarrow$ P2(y, z)
Relational Approaches
Relational Adaptive bootstrapping [Li et al., ACL-12]

**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

<table>
<thead>
<tr>
<th>Sentiment lexicon (camera)</th>
<th>Topic lexicon (camera)</th>
</tr>
</thead>
<tbody>
<tr>
<td>great, amazing, light</td>
<td>camera, product, screen,</td>
</tr>
<tr>
<td>recommend, excellent, etc.</td>
<td>photo, size, weight, quality,</td>
</tr>
<tr>
<td>artifacts, noise, never but,</td>
<td>price, memory, etc.</td>
</tr>
<tr>
<td>boring, etc.</td>
<td></td>
</tr>
</tbody>
</table>
Relational Approaches
Relational Adaptive bootstrapping (RAP) (cont.)

Reviews on cameras

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

Reviews on movies

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

In data level:
- Instance-based Approaches
- Feature-based Approaches
- Relational Approaches
- Parameter-based Approaches

In model level:
Some Advanced Research Issues in Transfer Learning

- How to transfer knowledge across heterogeneous feature spaces
- Active learning meets transfer learning
- Transfer learning from multiple sources
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 2009]
- [Quionero-Candela, et al., Data Shift in Machine Learning, MIT Press 2009]
- [Biltzer et al., Domain Adaptation with Structural Correspondence Learning, EMNLP 2006]
- [Pan et al., Cross-Domain Sentiment Classification via Spectral Feature Alignment, WWW 2010]
- [Pan et al., 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 via 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
Selected Applications of Transfer Learning

Qiang Yang and Sinno J. Pan

2013 PAKDD Tutorial
Brisbane, Australia
Part I. Cross Domain Transfer Learning for Activity Recognition


Demo

• Annotation
eHealth Demo

Sensor data
eHealth demo

Activity annotation
eHealth demo

Auto logging / activity recognition
(service in background)
Demo

• Recognition
eHealth demo

Real-time activity recognition
Demo

• **Profiling**
eHealth demo

Activity profiling
Activity profiling for health management
Key Problem: Recognizing Actions and Context (Locations)

AR: Activity Recognition via Sensors

- Walking?
- Buying Ticket?
- Open Door?

GPS and Other Sensors → Sensors → Sensors → Inferred through AR

Sightseeing → Watch show
1. Cross-Domain Activity Recognition

- **Challenge:**
  - Some activities without data (partially labeled)

- **Cross-domain activity recognition**
  - Use other activities with available labeled data

- Happen in kitchen
- Use cup, pot
- ...

Making coffee

Making tea
System Workflow

Source Domain
Labeled Data

Example:
\( \text{sim}(\text{“Make Coffee”}, \text{“Make Tea”}) = 0.6 \)

Target Domain
Pseudo Labeled
Data

Example: Pseudo Training Data: \(<\text{SS, “Make Tea”}, 0.6>\)

THE WEB

Weighted SVM Classifier

Example: <Sensor Reading, Activity Name> Example: <SS, “Make Coffee”>
Calculating Activity Similarities

• How similar are two activities?
  ◦ Use Web search results
  ◦ TFIDF: Traditional IR similarity metrics (cosine similarity)
  ◦ Example
    • Mined similarity between the activity “sweeping” and “vacuuming”, “making the bed”, “gardening”

Calculated Similarity with the activity "Sweeping"
Datasets: MIT PlaceLab

http://architecture.mit.edu/house_n/placelab.html

• MIT PlaceLab Dataset (PLIA2) [Intille et al. Pervasive 2005]
• Activities: Common household activities
Datasets: Intel Research Lab

- Intel Research Lab [Patterson, Fox, Kautz, Philipose, ISWC2005]
  - Activities Performed: 11 activities
  - Sensors
    - RFID Readers & Tags
  - Length:
    - 10 mornings

Picture excerpted from [Patterson, Fox, Kautz, Philipose, ISWC2005].
## Cross-Domain AR: Performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Accuracy with Cross Domain Transfer</th>
<th># Activities (Source Domain)</th>
<th># Activities (Target Domain)</th>
<th>Baseline (Random Guess)</th>
<th>Supervised (Upper bound)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Research Lab Dataset</td>
<td>63.2%</td>
<td>5</td>
<td>6</td>
<td>16.7%</td>
<td>78.3%</td>
</tr>
<tr>
<td>Amsterdam Dataset</td>
<td>65.8%</td>
<td>4</td>
<td>3</td>
<td>33.3%</td>
<td>72.3%</td>
</tr>
<tr>
<td>MIT Dataset (Cleaning to Laundry)</td>
<td>58.9%</td>
<td>13</td>
<td>8</td>
<td>12.5%</td>
<td>-</td>
</tr>
<tr>
<td>MIT Dataset (Cleaning to Dishwashing)</td>
<td>53.2%</td>
<td>13</td>
<td>7</td>
<td>14.3%</td>
<td>-</td>
</tr>
</tbody>
</table>

- Activities in the source domain and the target domain are generated from ten random trials, mean accuracies are reported.
Derek Hao Hu and Qiang Yang, IJCAI 2011

Transfer from Source Domain to Target Domain

\[ p(y_t \mid x_t) = \sum_{c^{(i)} \in L_s} p(c \mid x_t) \cdot p(y_t \mid c) \]
Proposed Approach

• Final goal: Estimate
  - We have
    \[ p(y_t|x_t) = \sum_{c(i) \in \mathcal{L}_s} p(c|x_t) \cdot p(y_t|c) \]
    
  - \( p(y_t|x_t) \approx p(\hat{c}|x_t) \cdot p(y_t|\hat{c}) \quad (\hat{c} = \arg\max_{c \in \mathcal{L}_s} p(c|x_t)) \) e:

  - Feature Transfer
  - Label Transfer
Experiments

• Datasets
  – UvA dataset [van Kasteren et al. Ubicomp 2008]
  – MIT Placelab (PLIA1) dataset [Intille et al. Ubicomp 2006]
  – Intel Research Lab dataset [Patterson et al. ISWC 2005]

• Baseline
  – Unsupervised Activity Recognition Algorithm [Wyatt et al. 2005]

• Different sensors for different datasets
Experiments:
Different Feature & Label Spaces

<table>
<thead>
<tr>
<th>K</th>
<th>MIT $\rightarrow$ UvA Acc(Var)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K = 5</td>
<td>59.8% (4.2%)</td>
</tr>
<tr>
<td>K = 10</td>
<td>57.5% (4.1%)</td>
</tr>
<tr>
<td>K = 15</td>
<td>51.0% (4.8%)</td>
</tr>
<tr>
<td>K = 20</td>
<td>41.0% (4.1%)</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>47.3% (4.1%)</td>
</tr>
</tbody>
</table>

Table 3: Algorithm performance of transferring knowledge from MIT PLIA1 to UvA dataset

<table>
<thead>
<tr>
<th>K</th>
<th>MIT $\rightarrow$ Intel Acc(Var)</th>
</tr>
</thead>
<tbody>
<tr>
<td>K = 5</td>
<td>60.5% (4.2%)</td>
</tr>
<tr>
<td>K = 10</td>
<td>61.2% (3.8%)</td>
</tr>
<tr>
<td>K = 15</td>
<td>53.2% (4.1%)</td>
</tr>
<tr>
<td>K = 20</td>
<td>42.0% (2.5%)</td>
</tr>
<tr>
<td>Unsupervised</td>
<td>42.8% (3.8%)</td>
</tr>
</tbody>
</table>

Table 4: Algorithm performance of transferring knowledge from MIT PLIA1 to Intel dataset

- Source: MIT PLIA1 dataset
- Target: UvA (Intel) datasets
Part II

• Source Free Transfer Learning

Source-Selection-free
Transfer Learning

Evan Xiang, Sinno Pan, Weike Pan, Jian Su, Qiang Yang
Transfer Learning

When we have some related source domains

Lack of labeled training data always happens

Supervised Learning

Transfer Learning
Where are the “right” source data?

We may have an extremely large number of choices of potential sources to use.
Outline of Source-Selection-Free Transfer Learning (SSFTL)

- **Stage 1: Building base models**
- **Stage 2: Label Bridging via Laplacian Graph Embedding**
- **Stage 3: Mapping the target instance using the base classifiers & the projection matrix**
- **Stage 4: Learning a matrix W to directly project the target instance to the latent space**
- **Stage 5: Making predictions for the incoming test data using W**
From the taxonomy of the online information source, we can “Compile” a lot of base classification models.
However, the label spaces of the based classification models and the target task can be different. The relationships between labels, e.g., similar or dissimilar, can be represented by the distance between their corresponding prototypes in the latent space, e.g., close to or far away from each other.
SSFTL – Mapping the target instance using the base classifiers & the projection matrix V

For each target instance, we can obtain a combined result on the label space via aggregating the predictions from all the base classifiers. However, do we need to recall the base classifiers during the prediction phase? The answer is No!

Target Instance

"Ipad2 is released in March, ..."

Then we can use the projection matrix V to transform such combined results from the label space to a latent space.
SSFTL – Learning a matrix $W$ to directly project the target instance to the latent space

For each target instance, we first aggregate its prediction on the base label space, and then project it onto the latent space.

$$V'^u F_S' = V' \sum_{i=1}^k \varepsilon_i F_{Si}'$$

$$\Omega_{D_T}'(W) = \frac{1}{n-\ell} \| W'X'^u - V'F'^u_S \|^2_F$$

$$\Omega_{D_T}(W) = \frac{1}{\ell} \| W'X - V_T'\phi(Y) \|^2_F$$

$$\min_W \Omega_{D_T}'(W) + \lambda_1 \| W \|^2_F + \lambda_2 \Omega_{D_T}(W)$$

Target Domain

Loss on unlabeled data

Loss on labeled data

Our regression model

Projection matrix
SSFTL – Making predictions for the incoming test data

Target Domain

Incoming Test Data

Learned Projection matrix

The learned projection matrix $W$ can be used to transform any target instance directly from the feature space to the latent space.

Projection matrix

$y^* = \arg \max_y P(y|x) = \frac{e^{-||W'x - v_y||_2^2}}{\sum_{y \in Y_T} e^{-||W'y - v_y||_2^2}}$

Therefore, we can make prediction directly for any incoming test data based on the distance to the label prototypes, without calling the base classification models.

HKUST - IJCAI 2011
Experiments - Datasets

- **Building Source Classifiers with Wikipedia**
  - 3M articles, 500K categories (mirror of Aug 2009)
  - 50,000 pairs of categories are sampled for source models

- **Building Label Graph with Delicious**
  - 800-day historical tagging log (Jan 2005 ~ March 2007)
  - 50M tagging logs of 200K tags on 5M Web pages

- **Benchmark Target Tasks**
  - 20 Newsgroups (190 tasks)
  - Google Snippets (28 tasks)
  - AOL Web queries (126 tasks)
  - AG Reuters corpus (10 tasks)
SSFTL - Building base classifiers Parallely using MapReduce

Input

Map

Reduce

The training data are replicated and assigned to different bins

If we need to build 50,000 base classifiers, it would take about two days if we run the training process on a single server. Therefore, we distributed the training process to a cluster with 30 cores using MapReduce, and finished the training within two hours.

These pre-trained source base classifiers are stored and reused for different incoming target tasks.

In each bin, the training data are paired for building binary base classifiers.
### Experiments - Results

**Table 1: Comparison results under varying numbers of labeled data in the target task (accuracy in %).**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>0</th>
<th>5</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RG</td>
<td>SSFTL</td>
<td>SVM</td>
<td>TSVM</td>
</tr>
<tr>
<td>20NG</td>
<td>50.0</td>
<td>80.3</td>
<td>69.8</td>
<td>75.7</td>
</tr>
<tr>
<td>Google</td>
<td>50.0</td>
<td>72.5</td>
<td>62.1</td>
<td>69.7</td>
</tr>
<tr>
<td>AOL</td>
<td>50.0</td>
<td>71.0</td>
<td>72.1</td>
<td>74.1</td>
</tr>
<tr>
<td>Reuters</td>
<td>50.0</td>
<td>72.7</td>
<td>69.7</td>
<td>63.3</td>
</tr>
</tbody>
</table>

- **Unsupervised SSFTL**
- **Semi-supervised SSFTL**

**Our regression model**

$$
\min_{W} \Omega_{D_T}(W) + \lambda_1 ||W||^2_F + \lambda_2 \Omega_{D_T}(W)
$$

**Parameter settings:**
- **Source models:** 5,000
- **Unlabeled target data:** 100%
- **lambda_2:** 0.01
Experiments - Results

Table 2: Comparison results on varying numbers of source classifiers (accuracy in %).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of source classifiers for SSFTTL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>250</td>
</tr>
<tr>
<td>20NG</td>
<td>76.3</td>
</tr>
<tr>
<td>Google</td>
<td>70.6</td>
</tr>
<tr>
<td>AOL</td>
<td>67.6</td>
</tr>
<tr>
<td>Reuters</td>
<td>72.2</td>
</tr>
</tbody>
</table>

For each target instance, we first aggregate its prediction on the base label space, and then project it onto the latent space.

\[ V'F^u_S = V' \sum_{i=1}^{k} \varepsilon_i F^u_{S_i} \]

**Parameter settings**
- Mode: Semi-supervised
- Labeled target data: 20
- Unlabeled target data: 100%
- \( \lambda_2: 0.01 \)

Our regression model:
\[
\min_W \Omega_{D_T}(W) + \lambda_1 \|W\|^2_F + \lambda_2 \Omega_{D_T}^u(W)
\]
# Experiments - Results

Table 3: Comparison results on varying size of unlabeled data in the target task (accuracy in %).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Unlabeled data involved in SSFTL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20%</td>
</tr>
<tr>
<td>20NG</td>
<td>80.5</td>
</tr>
<tr>
<td>Google</td>
<td>74.5</td>
</tr>
<tr>
<td>AOL</td>
<td>73.4</td>
</tr>
<tr>
<td>Reuters</td>
<td>75.5</td>
</tr>
</tbody>
</table>

Our regression model

$$\min_{\mathbf{W}} \Omega_{D_T^E}(\mathbf{W}) + \lambda_1 ||\mathbf{W}||_F^2 + \lambda_2 \Omega_{D_T^u}(\mathbf{W})$$

- Parameter settings:
  - Mode: Semi-supervised
  - Labeled target data: 20
  - Source models: 5,000
  - $\lambda_2$: 0.01
Experiments - Results

Table 4: Overall performance of SSFTL under varying values of $\lambda_2$ (accuracy in %).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\lambda_2$ of SSFTL</th>
<th>0</th>
<th>0.001</th>
<th>0.01</th>
<th>0.1</th>
<th>1</th>
<th>10</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>20NG</td>
<td></td>
<td>83.2</td>
<td>84.1</td>
<td>84.5</td>
<td><strong>85.3</strong></td>
<td>84.8</td>
<td>83.3</td>
<td>79.3</td>
</tr>
<tr>
<td>Google</td>
<td></td>
<td>76.6</td>
<td>79.1</td>
<td><strong>80.3</strong></td>
<td>78.7</td>
<td>78.2</td>
<td>77.4</td>
<td>74.3</td>
</tr>
<tr>
<td>AOL</td>
<td></td>
<td>78.3</td>
<td>79.5</td>
<td><strong>80.7</strong></td>
<td>79.1</td>
<td>78.8</td>
<td>76.3</td>
<td>73.4</td>
</tr>
<tr>
<td>Reuters</td>
<td></td>
<td>75.5</td>
<td>78.2</td>
<td><strong>80.1</strong></td>
<td>78.5</td>
<td>76.0</td>
<td>72.1</td>
<td>68.5</td>
</tr>
</tbody>
</table>

**Supervised SSFTL**

\[
\min_W \Omega_{D_T}^\prime(W) + \lambda_1 \|W\|_F^2 + \lambda_2 \Omega_{D_T}^\mu(W)
\]

**Semi-supervised SSFTL**

- Parameter settings:
  - Labeled target data: 20
  - Unlabeled target data: 100%
- Source models: 5,000

Our regression model

HKUST - IJCAI 2011
Experiments - Results

Table 5: Analysis on weighted and uniform SSFTTL under varying number of labeled data (accuracy in %).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Uniform SSFTTL</th>
<th>Weighted SSFTTL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>20NG</td>
<td>72.8</td>
<td>80.7</td>
</tr>
<tr>
<td>Google</td>
<td>64.1</td>
<td>67.0</td>
</tr>
<tr>
<td>AOL</td>
<td>69.8</td>
<td>71.7</td>
</tr>
<tr>
<td>Reuters</td>
<td>69.7</td>
<td>70.3</td>
</tr>
</tbody>
</table>

For each target instance, we first aggregate its prediction on the base label space, and then project it onto the latent space.

$$V'F^u_S = V' \sum_{i=1}^{k} \xi_i F^u_{S_i}$$

Loss on unlabeled data

$$\Omega_D^u_T(W) = \frac{1}{n-\ell} \|W'X^u - V'F^u_S \|_F^2$$

Our regression model

$$\min_W \Omega_D^u_T(W) + \lambda_1 \|W\|_F^2 + \lambda_2 \Omega_D^u_T(W)$$

-Parameter setttings-
  - Mode: Semi-supervised
  - Labeled target data: 20
  - Source models: 5,000
  - Unlabeled target data: 100%
  - lambda_2: 0.01
## Related Works

### Table 6: Summary of some related transfer learning works.

<table>
<thead>
<tr>
<th>Transfer learning methods</th>
<th>Scalability</th>
<th>Diff. label</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSP [Shi et al., 2009]</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>EigenTransfer [Dai et al., 2009]</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>MTL-MI [Quadrianto et al., 2010]</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>DAM [Duan et al., 2009]</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>LWE [Gao et al., 2008]</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>SSFTL</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Conclusion

- **Source-Selection-Free Transfer Learning**
  - When the potential auxiliary data is embedded in very large online information sources

- **No need for task-specific source-domain data**
  - We compile the label sets into a graph Laplacian for automatic label bridging

- **SSFTL is highly scalable**
  - Processing of the online information source can be done offline and reused for different tasks.
Advance Research Topics in Transfer Learning

Wei Fan

Huawei Noah's Ark Research Lab, Hong Kong
Predictive Modeling with Heterogeneous Sources

Xiaoxiao Shi   Qi Liu   Wei Fan
Qiang Yang   Philip S. Yu
Why learning with heterogeneous sources?

Standard Supervised Learning

Training (labeled)

Classifier

Test (unlabeled)

85.5%

New York Times

New York Times
Why heterogeneous sources?

In Reality…

Training (labeled)

Test (unlabeled)

Labeled data are insufficient!

How to improve the performance?

New York Times 47.3%
Why heterogeneous sources?

Labeled data from other sources

Target domain test (unlabeled)

1. Different distributions
2. Different outputs
3. Different feature spaces

47.3%

Reuter

New York Times
Real world examples

• Social Network:
  – Can various bookmarking systems help predict social tags for a new system given that their outputs (social tags) and data (documents) are different?

Wikipedia  ODP  Backflip  Blink  ……
Real world examples

- Applied Sociology:
  - Can the suburban housing price census data help predict the downtown housing prices?

<table>
<thead>
<tr>
<th>#rooms</th>
<th>#bathrooms</th>
<th>#windows</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
<td>12</td>
<td>XXX</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>11</td>
<td>XXX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#rooms</th>
<th>#bathrooms</th>
<th>#windows</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>XXXXX</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>5</td>
<td>XXXXX</td>
</tr>
</tbody>
</table>
Other examples

• Bioinformatics
  – Previous years’ flu data $\rightarrow$ new swine flu
  – Drug efficacy data against breast cancer $\rightarrow$
    drug data against lung cancer
  – ……

• Intrusion detection
  – Existing types of intrusions $\rightarrow$ unknown
    types of intrusions

• Sentiment analysis
  – Review from SDM $\rightarrow$ Review from KDD
Learning with Heterogeneous Sources

• The paper mainly attacks two sub-problems:
  – Heterogeneous data distributions
    • Clustering based KL divergence and a corresponding sampling technique
  – Heterogeneous outputs (to regression problem)
    • Unifying outputs via preserving similarity.
Learning with Heterogeneous Sources

• General Framework

Source data

Unifying data distributions

Unifying outputs

Target data

Source data

Target data
Unifying Data Distributions

• Basic idea:
  – Combine the source and target data and perform clustering.
  – Select the clusters in which the target and source data are similarly distributed, evaluated by KL divergence.
An Example

Combined Data

\[
KL_c(T|D) = \frac{2}{|T|} U + \log \frac{|D|}{|T|}
\]

\[
U = \sum_C \left( \frac{|T \cap C|^2}{|C|} \log \frac{|T \cap C|}{|D \cap C|} \right)
\]

|T| = 7
|D| = 8

|T \cap C_1| = 4
|D \cap C_1| = 5

|T \cap C_2| = 3
|D \cap C_2| = 2

C_1

C_2

C_3
Unifying Outputs

• Basic idea:
  – Generate initial outputs according to the regression model
  – For the instances similar in the original output space, make their new outputs closer.
Experiment

• Bioinformatics data set:

Table 1: Description of the data sets (#Feature =161)

<table>
<thead>
<tr>
<th>Order</th>
<th>Type</th>
<th>Size</th>
<th>Scale</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Regression</td>
<td>2431</td>
<td>0～99.99</td>
<td>[8]</td>
</tr>
<tr>
<td>2</td>
<td>Regression</td>
<td>561</td>
<td>1～127.8</td>
<td>[8]</td>
</tr>
<tr>
<td>3</td>
<td>Regression</td>
<td>601</td>
<td>0～100</td>
<td>[8]</td>
</tr>
<tr>
<td>4</td>
<td>Regression</td>
<td>290</td>
<td>2.1～98</td>
<td>[15]</td>
</tr>
<tr>
<td>5</td>
<td>Regression</td>
<td>344</td>
<td>0.2～98.5</td>
<td>[15]</td>
</tr>
<tr>
<td>6</td>
<td>Classification</td>
<td>7443</td>
<td>4 classes</td>
<td>[10]</td>
</tr>
<tr>
<td>7</td>
<td>Classification</td>
<td>196</td>
<td>2 classes</td>
<td>[16]</td>
</tr>
</tbody>
</table>

Note: Some references, such as [8], refer to several data sets from different research groups.
Experiment
Experiment

• Applied sociology data set:

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton</td>
<td>18</td>
<td>2.47~21.46</td>
</tr>
<tr>
<td>Boston Roxbury</td>
<td>19</td>
<td>12.03~36.98</td>
</tr>
<tr>
<td>Lynn</td>
<td>22</td>
<td>6.58~27.71</td>
</tr>
<tr>
<td>Boston Savin Hill</td>
<td>23</td>
<td>15.17~34.02</td>
</tr>
<tr>
<td>Cambridge</td>
<td>30</td>
<td>1.73~29.53</td>
</tr>
<tr>
<td>Somerville</td>
<td>15</td>
<td>11.12~34.41</td>
</tr>
<tr>
<td>South Boston</td>
<td>10</td>
<td>3.53~18.46</td>
</tr>
<tr>
<td>Brookline</td>
<td>11</td>
<td>7.67~18.66</td>
</tr>
<tr>
<td>East Boston</td>
<td>11</td>
<td>10.29~19.01</td>
</tr>
<tr>
<td>Quincy</td>
<td>11</td>
<td>9.38~29.55</td>
</tr>
</tbody>
</table>
Experiment

(a) Newton

(b) Boston Roxbury

(c) Lynn

(d) Boston Savin Hill

(e) Cambridge
Conclusions

• Problem: Learning with Heterogeneous Sources:
  • Heterogeneous data distributions
  • Heterogeneous outputs
• Solution:
  • Clustering based KL divergence help perform sampling
  • Similarity preserving output generation help unify outputs
Transfer Learning on Heterogeneous Feature Spaces via Spectral Transformation

Xiaoxiao Shi, Qi Liu, Wei Fan, Philip S. Yu, and Ruixin Zhu
Motivation

Standard Supervised Learning

Training documents (labeled)

Classifier

Test documents (unlabeled)

The New York Times

85.5%

The New York Times
In Reality…

How to improve the performance?

Labeled data are insufficient!

Huge set of unlabeled documents

The New York Times 47.3%
Learning Formulations

Supervised Learning

Unsupervised Learning

Semi-supervised Learning

Transfer Learning
Labeled data from other sources

Target domain test (unlabeled)

Heterogeneous datasets:
1. Different data distributions: $P(x_{train})$ and $P(x_{test})$ are different
2. Different outputs: $y_{train}$ and $y_{test}$ are different
3. Different feature spaces: $x_{train}$ and $x_{test}$ are different
• WiFi-based localization tracking [Pan et al'08]
• Collaborative Filtering [Pan et al'10]
• Activity Recognition [Zheng et al'09]
• Text Classification [Dai et al'07]
• Sentiment Classification [Blitzer et al ‘07]
• Image Categorization [Shi et al’10]
• .......
Issues

- Different data distributions: $P(x_{\text{train}})$ and $P(x_{\text{test}})$ are different

- Chicago Tribune focuses more on Chicago local news

- Reuters focuses more on global news

- Wikipedia focuses more on scientific/objective documents
Issues

• Different outputs: \( y_{\text{train}} \) and \( y_{\text{test}} \) are different
Issues

• Different feature spaces (the focus on the paper)
  – Drug efficacy tests:
    • Physical properties
    • Topological properties

– Image Classification
  • Wavelet features
  • Color histogram
Unify different feature spaces

• Different number of features; different meanings of the features, no common feature, no overlap.

• Projection-based approach HeMap
  – Find a projected space where (1) the source and target data are similar in distribution; (2) the original
Unify different feature spaces via HeMap

Optimization objective of HeMap:

\[
\min_{\mathbf{B}_T, \mathbf{B}_S} \ell(\mathbf{B}_T, \mathbf{T}) + \ell(\mathbf{B}_S, \mathbf{S}) + \beta \cdot D(\mathbf{B}_T, \mathbf{B}_S) \quad (1)
\]

\[
\ell(\mathbf{B}_T, \mathbf{T}) = \| \mathbf{B} \ell(\mathbf{B}_S, \mathbf{S}) = \tilde{D}(\mathbf{B}_T, \mathbf{B}_S) = \frac{1}{2} (\ell(\mathbf{B}_T, \mathbf{S}) + \ell(\mathbf{B}_S, \mathbf{T}))
\]

The linear projection error
The linear projection error
The difference between the projected data

where \( \mathbf{B}_T \in \mathbb{R}^{r \times k}, \mathbf{B}_S \in \mathbb{R}^{q \times k} \) are the projected matrices of \( \mathbf{T} \) and \( \mathbf{S} \) respectively.
Unify different feature spaces via HeMap

With some derivations, the objective can be reformulated as (more details can be found in the paper):

\[
\text{Theorem 1: The minimization problem in Eq. (4) is equivalent to the following maximization problem:}
\]

\[
\min_{B_T B_T = I, B_S B_S = I} G = \max_{B^T B = I} \text{tr}(B^T AB) \quad (6)
\]

where

\[
B = \begin{bmatrix} B_T \\ B_S \end{bmatrix}, \quad A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}. \quad (7)
\]

\[
A_1 = 2TT^T + \frac{\beta^2}{2} SS^T, \quad A_4 = \frac{\beta^2}{2} TT^T + 2SS^T
\]

\[
A_2 = A_3^T = \beta (SS^T + TT^T)
\]
Algorithm flow of HeMap

Construct matrix $A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}$

$A_1 = 2TT^T + \frac{\beta^2}{2} SS^T$, $A_4 = \frac{\beta^2}{2} TT^T + 2SS^T$

Calculate the top-$k$ eigenvalues of $A$, and their corresponding eigenvectors $U = [u_1, \cdots, u_k]$.

$B_T$ is the first half rows of $U$; $B_S$ is the second half rows of $U$. 
Generalized HeMap to handle heterogeneous data (different distributions, outputs and feature spaces)

If the selected source data are limited (e.g., 1%), claim “too risky to use the source”

otherwise, apply selected source data as new training data
Unify different distributions and outputs

• Unify different distributions
  – Clustering based sample selection [Shi et al, 09]

• Unify different outputs
  Bayesian like schema

\[ p(y|x) = \sum_v (p(v|x)p(y|v)) \]  

where \( x \) is the data to be predicted; \( y \) is the target label; and \( v \) denotes the output from the source task.
Theorem 4: Let $\mathcal{H}$ be a hypothesis space. Let $T$ be unlabeled samples of size $r$. Let $S$ be a labeled sample of size $q$ generated by drawing $\vartheta q$ points from target data and $(1 - \vartheta)q$ points from source data. If $\hat{h} \in \mathcal{H}$ is the empirical minimizer of the error on $S$ and $h^* = \min_{h \in \mathcal{H}} \epsilon(h)$ is the target risk minimizer, then with probability at least $1 - \delta$ (over the choice of the samples),

$$
\epsilon(\hat{h}) \leq \epsilon(h^*) + 2 \sqrt{\frac{\alpha^2}{\beta} + \frac{(1 - \alpha)^2}{1 - \beta}} \sqrt{\frac{g(\hat{h}) \log(2q) - \log \delta}{2q}} \\
+ 2(1 - \alpha) \left( \frac{1}{2} d(T, S) + 4 \sqrt{\frac{2g(\hat{h}) \log r + \log \frac{4}{\delta}}{r}} + \xi \right)
$$

$\alpha$ and $\beta$ are domain-specific parameters; $g(\hat{h})$ is model complexity.

Principle I: minimize the difference between target and source datasets

Principle II: minimize the combined expected error by maintaining the original structure (minimize projection error)
Experiments

• Drug efficacy prediction
  – The dataset is collected by the College of Life Science and Biotechnology of Tongji University, China. It is to predict the efficacy of drug compounds against certain cell lines.
  – The data are generated in two different feature spaces
    • general descriptors: refer to physical properties of compounds
    • drug-like index: refer to simple topological indices of compounds.
Experiments

(a) Target is data set 1; source is data set 2
(b) Target is data set 2; source is data set 1
(c) Target is data set 3; source is data set 4
(d) Target is data set 4; source is data set 3
(e) Target is data set 5; source is data set 6
(f) Target is data set 6; source is data set 5
(g) Target is data set 7; source is data set 8
(h) Target is data set 8; source is data set 7
Experiments

- Image classification

Cartman & Bonsai

Homer Simpson & Cactus

Homer Simpson & Coin

Superman & CD
Experiments

(a) Target is Cartman and Bonsai; source is Homer Simpson and Cactus
(b) Target is Homer Simpson and Cactus; source is Cartman and Bonsai
(c) Target is Homer Simpson and Coin; source is Superman and CD
(d) Target is Superman and CD; source is Homer Simpson and Coin
Conclusions

• Extends the applicability of supervised learning, semi-supervised learning and transfer learning by using heterogeneous data:
  – Different data distributions
  – Different outputs
  – Different feature spaces

• Unify different feature spaces via linear projection with two principles
  – Maintain the original structure of the data
  – Maximize the similarity of the two data in the projected space
Cross Validation Framework to Choose Amongst Models and Datasets for Transfer Learning

Erheng Zhong¶, Wei Fan‡, Qiang Yang¶, Olivier Verscheure‡, Jiangtao Ren†
Transfer Learning: What is it

Definition

“source-domains” to improve “target-domain”: short of labeled information.

1. WiFi-based localization tracking [Pan et al'08]
2. Collaborative Filtering [Pan et al'10]
3. Activity Recognition [Zheng et al'09]
4. Text Classification [Dai et al'07]
5. Sentiment Classification [Blitzer et al'07]
6. Image Categorization [Shi et al'10]
Application

Indoor WiFi localization tracking

(a) WiFi signal at time period 1

(b) WiFi signal at time period 2

(Lx, Ly) is the coordinate of location.
Application

Collaborative Filtering

\[ R^{(1)} \text{ Book} \]

\[ <\text{User, Item, Rating}> \text{ Matrix} \]

\[ R^{(2)} \text{ Music} \]

\[ \text{Movie} \]

\[ R_{(target)} \]
Transfer Learning: How it works

Data Selection
- Limited Labeled Data from Target-domain
- Lots of Labeled Data from Source-domain

Model Selection
- Algorithm and parameters

Adaptation

Predict
- Trained Model
- Unlabeled Data from Target-domain
Re-cast: Model and Data Selection

(1) How to select the right transfer learning algorithms?

(2) How to tune the optimal parameters?

(3) How to choose the most helpful source-domain from a large pool of datasets?
Model & Data Selection  Traditional Methods

1. Analytical techniques: AIC, BIC, SRM, etc.

\[
\hat{f} = \arg \min_{f} \frac{1}{n} \sum_{x \in X_s} \left| P_s(y|x) - P(y|x, f) \right| + \Theta_f
\]

2. k-fold cross validation

\[
\hat{f} = \arg \min_{f} \frac{1}{k} \sum_{j=1}^{k} \sum_{(x,y) \in S_j} \left| P_s(y|x) - P(y|x, f_j) \right|
\]
Model & Data Selection

Issues

1. $P_s(x) \neq P_t(x)$

   The estimation is not consistent: $\lim_{n \to \infty} (\hat{f}) \neq f^*$

   Ideal Hypothesis:
   
   $$f^* = \arg\min_f \mathbb{E}_{x \sim P_t(x)} \left| P_t(y \mid x) - P(y \mid x, f) \right| + \Theta_f$$

2. $P_s(y \mid x) \neq P_t(y \mid x)$

   A model approximating $P_s(y \mid x)$ is not necessarily close to $P_t(y \mid x)$

The number of labeled data in target domain is limited and thus the directly estimation of $P_t(y \mid x)$ is not reliable.
Model & Data Selection

Model Selection Example

If we choose the wrong model....
Model & Data Selection

Data Selection Example

If we choose the wrong source-domain....
Transfer Cross-Validation (TrCV)

New criterion for transfer learning

$$\hat{f} = \arg \min_{f} \frac{1}{n} \sum_{x \in X_s} \frac{P_t(x)}{P_s(x)} \left| P_t(y|x) - P(y|x, f) \right|$$

Hard to calculate in practice

1. The density ratio between two domains
2. The difference between the conditional distribution estimated by model $f$ and the true conditional distribution.

Reverse Validation

How to calculate this difference with limited labeled data?

Practical method: Transfer Cross-Validation (TrCV)

$$\hat{f} = \arg \min_{f} \frac{1}{k} \sum_{j=1}^{k} \sum_{(x, y) \in S_j} \frac{P_t(x)}{P_s(x)} \left| P_t(y|x) - P(y|x, f) \right|$$

Density Ratio Weighting
Density Ratio Weighting

- The selected model is an unbiased estimator to the ideal model $f^*$.

**Lemma 1.** $\ell_w(\hat{f}) + \Theta_{\hat{f}} = \ell^*(f^*) + \Theta_{f^*}$, when $n \to \infty$ and $f^*$ and $\hat{f}$ belong to the same hypothesis class.

$\ell^*(f^*)$ is the expected loss to approximate $P_t(y|x)$

$$\ell_w(\hat{f}) = \frac{1}{n} \sum_{x \in X_s} \frac{P_t(x)}{P_s(x)} \left| P_t(y|x) - P(y|x, \hat{f}) \right|$$

$\Theta_f$ is the model complexity

Important property to choose the right model even when $P(x)$ and $P(y|x)$ are different.

- We adopt an existing method KMM (Huang et al’07) for density ratio weighting.
- Reverse Validation to estimate $P_t(y|x) - P(y|x, f)$ (next slide)
Reverse Validation

- $S_i$: The source-domain data in i-th fold
- $\overline{S}_i$: The remaining data
- $\overline{Y}_u^i$: The predicted label of $X_u$ in i-th fold
- $\overline{Y}_s^i$: The predicted label of $S_i$ in i-th fold
- $Y_s^i$: The true label of $S_i$ in i-th fold
- $X_u$, $X_\ell$: The unlabeled and labeled target-domain data
Properties

- The selected model is an unbiased estimator to the ideal one. [Lemma 1]
- The model selected by the proposed method has a generalization bound over target-domain data. [Theorem 1]
- The value of reverse validation $r(x)$ is related to the difference between true conditional probability and model approximation $|P(y|x, f_i) - P_t(y|x)|$
- The confidence of TrCV has a bound.

$$Pr \left\{ -z < \frac{\varepsilon_u(f) - \varepsilon(f)}{\sqrt{\varepsilon(f)} \cdot (1 - \varepsilon(f))/n} < z \right\} \approx \lambda$$

- $\varepsilon_u(f)$: the accuracy estimated by TrCV
- $\varepsilon(f)$: the true accuracy of $f$
- $z$: $(1 + \lambda)/2$-th quantile point of the standard normal distribution
Experiment Data Set

- Wine Quality: two subsets related to red and white variants of the Portuguese “Vinho Verde” wine.

| Data Set         | $|S|$ | $|T|$ | Description               |
|------------------|-----|-----|---------------------------|
| Red-White (RW)   | 1599| 4998| physicochemical variables |
| White-Red (WR)   | 4998| 1599|                           |

For algorithm and parameters selection
**Experiment**  \[\textbf{Data Set}\]

- Reuters-21578: the primary benchmark of text categorization formed by different news with a hierarchical structure.

| Data Set               | $|S|$  | $|T|$  | Description                      |
|------------------------|-------|-------|----------------------------------|
| orgs vs. people (ope)   | 1016  | 1046  | Documents from different subcategories |
| orgs vs. places (opl)   | 1079  | 1080  |                                   |
| people vs. places (pp)  | 1239  | 1210  |                                   |

For algorithm and parameters selection
• SyskillWebert: the standard dataset used to test web page ratings, generated by the HTML source of web pages plus the user rating. We randomly reserve “Bands-recording artists” as source-domain and the three others as target-domain data.

| Data Set            | $|S|$ | $|T|$ | Description                        |
|---------------------|-----|-----|-----------------------------------|
| Sheep(Sp)           | 61  | 65  | Web pages with different contents |
| Biomedical(BI)      | 61  | 131 |                                    |
| Goats(Gs)           | 61  | 70  |                                    |

For algorithm and parameters selection
### Experiment Data Set

- **20-Newsgroup**: primary benchmark of text categorization similar to Reuters-21578

<table>
<thead>
<tr>
<th>Data Set</th>
<th>S</th>
<th>T</th>
<th></th>
<th>S</th>
<th></th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>comp</td>
<td>windows vs. motorcycles</td>
<td>graphics</td>
<td>1596</td>
<td>1969</td>
<td>1957</td>
<td></td>
</tr>
<tr>
<td>vs.</td>
<td>pc.hardware vs. baseball</td>
<td>vs. autos</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rec</td>
<td>mac.hardware vs. hockey</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sci</td>
<td>crypt vs. guns</td>
<td>electronics</td>
<td>1895</td>
<td>1761</td>
<td>1924</td>
<td></td>
</tr>
<tr>
<td>vs.</td>
<td>med vs. misc</td>
<td>vs. mideast</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>talk</td>
<td>space vs. religion</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For source-domain selection
Experiment  
Baseline methods

- **SCV**: standard k-fold CV on source-domain
- **TCV**: standard k-fold CV on labeled data from target-domain
- **STV**: building a model on the source-domain data and validating it on labeled target-domain data
- **WCV**: using density ratio weighting to reduce the difference of marginal distribution between two domains, but ignoring the difference in conditional probability.

\[
\hat{f} = \arg\min_{f} \frac{1}{k} \sum_{j=1}^{k} \sum_{(x,y) \in S_j} \frac{P_t(x)}{P_s(x)} \left| \frac{P_s(y|x)}{P_s(x)} - P(y|x, f_j) \right|
\]
Experiment  Other settings

• Algorithms:
  – Naive Bayes (NB), SVM, C4.5, K-NN and NNge(Ng)
  – TrAdaBoost (TA): instances weighting [Dai et al.'07]
  – LatentMap (LM): feature transform [Xie et al.'09]
  – LWE: model weighting ensemble [Gao et al.'08]

• Evaluation: if one criterion can select the better model in the comparison, it gains a higher measure value.

\[ \text{corr} = C^2_{|\mathcal{H}|} - \sum_{f,g \in \mathcal{H}} \left[ \left( \varepsilon(f) - \varepsilon(g) \right) \times \left( v(f) - v(g) \right) < 0 \right] \]

\varepsilon(\cdot) \text{ and } v(\cdot) \quad \text{The accuracy and value of criteria (e.g. TrCV, SCV, etc)}

\quad C^2_{|\mathcal{H}|} \quad \text{The number of comparisons between models}
Results

Algorithm Selection

<table>
<thead>
<tr>
<th>Method</th>
<th>RW</th>
<th>WR</th>
<th>ope</th>
<th>opl</th>
<th>pp</th>
<th>Sp</th>
<th>Bl</th>
<th>Gs</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCV</td>
<td>18</td>
<td>17</td>
<td>13</td>
<td>17</td>
<td>13</td>
<td>19</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>TCV</td>
<td>17</td>
<td>18</td>
<td>14</td>
<td>17</td>
<td>10</td>
<td>15</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>STV</td>
<td>16</td>
<td>15</td>
<td>13</td>
<td>15</td>
<td>14</td>
<td>18</td>
<td>17</td>
<td>22</td>
</tr>
<tr>
<td>WCV</td>
<td>20</td>
<td>19</td>
<td>17</td>
<td>19</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>TrCV</td>
<td>22</td>
<td>23</td>
<td>22</td>
<td>20</td>
<td>22</td>
<td>22</td>
<td>20</td>
<td>20</td>
</tr>
</tbody>
</table>

6 win and 2 lose!
**Results**

**Parameter Tuning**

<table>
<thead>
<tr>
<th>Method</th>
<th>RW</th>
<th>WR</th>
<th>ope</th>
<th>op1</th>
<th>pp</th>
<th>Sp</th>
<th>Bl</th>
<th>Gs</th>
<th>RW</th>
<th>WR</th>
<th>ope</th>
<th>op1</th>
<th>pp</th>
<th>Sp</th>
<th>Bl</th>
<th>Gs</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCV</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>TCV</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>STV</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>8</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>3</td>
<td>8</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>WCV</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>TrCV</td>
<td>5</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>5</td>
<td>3</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

13 win and 3 lose!
## Results

Source-domain Selection

<table>
<thead>
<tr>
<th>Method</th>
<th>NB</th>
<th>SVM</th>
<th>C45</th>
<th>KNN</th>
<th>Ng</th>
<th>TA</th>
<th>LM</th>
<th>LWE</th>
<th>Pr</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCV</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>1</td>
<td>6</td>
<td>436</td>
</tr>
<tr>
<td>STV</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>371</td>
</tr>
<tr>
<td>TCV</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>399</td>
</tr>
<tr>
<td>WCV</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>6</td>
<td>442</td>
</tr>
<tr>
<td>TrCV</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>512</td>
</tr>
</tbody>
</table>

No lose!
Results  Parameter Analysis

TrCV achieves the highest correlation value under different number of folds from 5 to 30 with step size 5.
When only a few labeled data (< $0.4 \times |T|$) can be obtained in the target-domain, the performance of TrCV is much better than both SVT and TCV.
Conclusion

• Model and data selection when margin and conditional distributions are different between two domains.

• Key points
  – Point-1 Density weighting to reduce the difference between marginal distributions of two domains;
  – Point-2 Reverse validation to measure how well a model approximates the true conditional distribution of target-domain.

• Code and data available from the authors
  – www.weifan.info
Thank you!
Thanks!