Introduction to Transfer Learning

Jindong Wang
wangjindong@ict.ac.cn

Jun. 3, 2016
Content

Motivation
  Why Transfer Learning?

Introduction

Taxonomy
  Category
  Transfer Learning with Deep Learning
  Negative Transfer

Application
  Activity Recognition
  Other Applications
  Resources

Future Work

References
Activity Recognition: An Example

\(A: \text{labeled Wi-fi} \rightarrow C: \text{unlabeled Bluetooth, but how?}

Sentiment Classification: Another Example

Only sentiments on DVD, how to obtain those on Electronics?
Motivation
Why Transfer Learning?

Basics
- Building every model from scratch is time-consuming and expensive.
- But there are many existing knowledge. Can we reuse them?

(a) Traditional Machine Learning
(b) Transfer Learning
Introduction

Brief History

The Origin of TL

Thorndike and Woodworth in 1901: how individuals transfer in one context to another context that share similar characteristics [TW01].

Common Definition

Wikipedia: research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem [wik].

Proceedings

Data mining: ACM SIGKDD, IEEE ICDM, PKDD
Machine learning: ICML, NIPS, ECML, AAAI, IJCAI
Applications: ACM SIGIR, WWW, ACL, IEEE TKDE
Introduction
When can we Use TL?

Traditional ML Assumptions

▶ Training and testing samples must be in the same feature distributions.
▶ Training samples must be enough.

TL conditions

▶ Source and target domains do not need to be in the same distributions.
▶ Less training samples, even none.
▶ Example: getting labeled samples is time-consuming and expensive.
Basic notations

- **Domain**: $D = (X, P(X))$, $X$: feature space, $P(X)$: marginal distribution where $X = \{X_1, X_2, \cdots, X_n\}$
- **Task**: $T = (Y, f(\cdot))$, $Y$: label space, $f(\cdot)$: objective predictive function.

Transfer learning

- **Source domain**: $D_S = \{X_S, P(X_S)\}$
- **Source task**: $T_S = \{Y_S, f_S(\cdot)\}$
- **Target domain**: $D_T = \{X_T, P(X_T)\}$
- **Target task**: $T_T = \{Y_T, f_T(\cdot)\}$
- **Goal**: $\min \epsilon(f_T(X_T), Y_T)$
- **Conditions**: $D_T \neq D_S$ or $T_T \neq T_S$ with $(D_T, D_S, Y_T, Y_S)$ may be unknown, respectively
Taxonomy
Different Views

By Data Distribution
- Inductive TL
- Transductive TL
- Unsupervised TL

By Methodology
- Instance based TL
- Feature based TL
- Parameter/model based TL
- Relational TL

<table>
<thead>
<tr>
<th>Transfer Learning Settings</th>
<th>Related Areas</th>
<th>Source Domain Labels</th>
<th>Target Domain Labels</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inductive Transfer Learning</td>
<td>Multi-task Learning</td>
<td>Available</td>
<td>Available</td>
<td>Regression, Classification</td>
</tr>
<tr>
<td></td>
<td>Self-taught Learning</td>
<td>Unavailable</td>
<td>Available</td>
<td>Regression, Classification</td>
</tr>
<tr>
<td>Transductive Transfer Learning</td>
<td>Domain Adaptation, Sample Selection Bias, Co-variate Shift</td>
<td>Available</td>
<td>Unavailable</td>
<td>Regression, Classification</td>
</tr>
<tr>
<td>Unsupervised Transfer Learning</td>
<td>Unavailable</td>
<td>Unavailable</td>
<td>Unavailable</td>
<td>Clustering, Dimensionality Reduction</td>
</tr>
</tbody>
</table>

Figure: Transfer learning settings[PY10]
### Taxonomy
Categorized by Domain and Task

#### Inductive transfer learning
Given \( T_S \neq T_T \) under conditions:
- A lot of labeled \( D_S \) or
- No labeled \( D_S \)

#### Transductive transfer learning
Given \( T_S = T_T \) under conditions:
- \( X_S \neq X_T \) or
- \( X_S = X_T \) and \( P(X_S) \neq P(X_T) \)

#### Unsupervised transfer learning
Given \( T_S \neq T_T \) under conditions:
- No labeled \( D_S \) and \( D_T \)
<table>
<thead>
<tr>
<th>Taxonomy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Categorized by Approaches</strong></td>
</tr>
<tr>
<td>Instance based transfer learning</td>
</tr>
<tr>
<td>Reuse source domain: instance re-weighting and importance sampling</td>
</tr>
<tr>
<td>Feature based transfer learning</td>
</tr>
<tr>
<td>Learn good feature representation of target domain</td>
</tr>
<tr>
<td>Parameter based transfer learning</td>
</tr>
<tr>
<td>Transfer models between source and target domains</td>
</tr>
<tr>
<td>Relational transfer learning</td>
</tr>
<tr>
<td>Relationships are same in source and target domains</td>
</tr>
</tbody>
</table>
Transfer Learning with Deep Learning
Comparison

Deep Learning: Nonlinear Representations
- Hierarchical network.
- disentangle different explanatory factors of variation behind data samples.

Transfer Learning: Alleviation
- Doesn’t need a large amount of data.
Applying transfer learning with deep learning outperforms directly applying Deep Learning.

Unsupervised domain adaptation Amazon → Webcam over time

- TL without DL
- TL with DL
- DL without TL
**Negative Transfer**

Negative transfer happens when source domain data and task contribute to **reduced** performance of learning in the target domain.

**Negative Transfer Conditions**

- Domains are too dissimilar [RMKD05]
- Conditional Kolmogorov complexity is not related [BH03]
- Tasks are not well-related [BH03]

**Transitive** transfer learning [TSZY15]

**Figure:** TTL tries to bridge the source and target domain using auxiliary sources
Can we transfer from existing activities domain to a different but related domain? [ZHY09]

**Problem Formulation**

- Labeled source activities: \( A_{src} = \{a_1, a_2, \ldots, a_m\} \)
- Unlabeled target activities: \( A_{tar} = \{a_{m+1}, a_{m+2}, \ldots, a_n\} \), \( A_{src} \cap A_{tar} = \emptyset \).

**Figure:** Cross domain activity recognition
Similarity Measure

What is a good similarity measure?

**Maximum Mean Discrepancy [BGR+06]**

- \( X = \{x_1, x_2, \ldots, x_n\}, Y = \{y_1, y_2, \ldots, y_m\}, \text{i.i.d} \)
- \( \| \cdot \|_\mathcal{H} \): Reproducing Hilbert Kernel Space
- \( \phi \): kernel function, like Gaussian

\[
MMD^2(X,Y) = \| \frac{1}{n} \sum_{i=1}^{n} \phi(x_i) - \frac{1}{m} \sum_{i=1}^{m} \phi(y_i) \|_\mathcal{H}^2
\]

**Other Measures**

- **Cosine similarity**: \( \text{sim}(X,Y) = \frac{X \cdot Y}{\|X\| \|Y\|} \)
- **Kullback-Leibler (KL) divergence**: \( D_{KL}(P\|Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \)
- **Jensen-Shannon divergence (JSD)**: let \( M = \frac{1}{2}(P + Q) \), then
  \[
  JSD(P\|Q) = \frac{1}{2} D_{KL}(P\|M) + \frac{1}{2} D_{KL}(Q\|M)
  \]
## Applications I

**Other Applications**

### Text Mining
- Unified clustering and shared knowledge transfer [DXYY07a]
- Transfered Bayes [DXYY07b]

### Image Processing
- Text to image clustering [DCX+08]
- Heterogeneous transfer learning [ZCL+11b]

### Collaborative Filtering
- Sub feature space transfer [PXY12]
- Latent feature sharing [CLY10]
### Indoor Localization
- Transfer similar floors [WZZY10]
- A transfer learning framework [ZY14]

### Activity Recognition
- Cross-people mobile phone based AR [ZCL$^+$11a]
- Community similarity network [LXL$^+$11]
Applications
Resources

Resources

- Open source program: http://www.cse.ust.hk/TL/
- Sinno Jialin Pan: http://www.ntu.edu.sg/home/sinnopan/
- Wenyuan Dai: http://www.4paradigm.com/homepage.html

Survey

- A survey on Transfer Learning [PY10].
- A survey of Transfer Learning [WKW16].
- Transfer learning for activity recognition: A survey [CFK13].
- Fuzzy Transfer Learning: Methodology and application [SC15].
Future Work

- Reliable similarity measure
- Transfer within different algorithms
- More accurate theoretical support

Figure: The future of machine learning [Yan16]


Thank you for your listening