Domain Adaptation in Real World Applications

Presented by: Ivor W. Tsang
Nanyang Technological University
Outline

1. Review of Domain Adaptation (DA)
2. Domain Selection and Multi-Source Learning
   - Healing Source Bias via Source Classifier Selection
   - DA via Distance Metric Selection
3. Optimizing Performance Measures for DA
4. DA in Heterogeneous Feature Domains
5. Conclusion
Domain Adaptation (DA)

• Task
  ◦ Learning in a new domain of interest with only a few or even no labeled training data

• Paradigm
  ◦ To retain and apply previous knowledge learned from one or multiple existing domains (a.k.a., the source domains) to improve learning in the new domain (a.k.a., the target domain).
Sentiment Applications

- Positive or negative rating

Examples:
- Amazon products
- Movie
- Any new product to be launched soon
  - To get the feeling of what the customers feels
  - E.g. IPhone 5, Happy Feet two
Sentiment Prediction

**Labeled Data**
- Review 1
- Review 2
- ... 
- Review n

**Feature Extractions**
- Token frequency features
- TF-IDF features

**Prediction**
- Review A
- Review B
- Review C

**Classifiers Building**
- Learning methods: SVMs, kNN, AdaBoost, etc.
Sentiment Prediction

• Labeled data are difficult to collect, especially when a new product is launched

• How to get sufficient labeled data for training robust classifiers?
Sentiments from Amazon

- **Target task**
  - Kitchen appliances

- **Source domains (Described with same language)**
  - Book
  - DVDs
  - Electronics

- Can we leverage existing labeled data from other source domains?
Visual Concept Detection

Labeled Photos/Videos

Feature Extractions
- Global features: color moment, Gabor texture, etc.
- Local features: SIFT and other local features

Classifiers Building
- Learning methods: SVMs, kNN, AdaBoost, etc.

Prediction
Visual Concept Detection

- **Current Task**
  - To label a set of unlabeled photos in personal albums (*Target* domain)
  - **No** or few labeled photos
    - Calibration is very labor intensive
    - Unable to obtain label information for *rare concepts*

- **Many Available Source Domains**
  - Lots of labeled data from *Internet*
    - e.g. photos with text descriptions from *Flickr.com*
    - e.g. videos with text descriptions from *Youtube*
    - e.g. named face images from *Facebook*
  - Which sources can help label target unlabeled photos?
Photos from Different Domains

Examples are taken from Saenko
Videos from Different Domains

- **Data distribution mismatch** between consumer videos and web videos
  - Consumer videos: Naturally captured
  - Web videos: Compressed and edited
Assumptions in Traditional MLs

- Training and test data (e.g. images, videos) come from the same task and the same domain
  - Generally, test data follow the same distribution of the extracted feature space as that of training data

- The learned model (e.g. SVM classifier)
  - is usually learned from scratch, without using prior models of similar learning tasks
  - is directly applied to test data of the same prediction tasks
Marginal and Predictive Distributions

- Source and target domains may have:
  - Different marginal distribution
    \[ P(x), \text{ where } x = \{x_1, x_2, \ldots, x_n\} \in X \]
  - Different predictive distribution
    \[ P(Y|x), \text{ where } Y = \{y_1, y_2, \ldots, y_n\} \text{ and } y_i \in Y \]
Existing DA Approaches

• **Instance Re-weighting**
  - Kernel Mean Matching (KMM) [Huang et al, NIPS’06]
  - Covariate Shift Adaptation [Sugiyama et al, JMLR’07]

• **Feature Weighting**
  - Structural Correspondence Learning (SCL) [Blitzer et al, EMNLP’06]
  - Feature Augmentation (FA) [Daume III, ACL’07]
  - Transfer Component Analysis (TCA) [Pan et al, IJCAI’09, TNN’11]
  - Domain Transfer Multiple Kernel Learning [Duan et al, CVPR’09, TPAMI’12]

• **Model / Classifier Adaptation**
  - Adaptive SVM (ASVM) [Yang et al, ACM Multimedia’07]
  - Multiple Convex Combination (MCC) [Schweikert et al, NIPS’09]
  - Domain Adaptation Machine (DAM) [Duan et al, ICML’09, TNNLS’12]
Limitations and Challenges in DA

- These DA methods typically assume the source and target domains having the same predictive distribution (i.e., $P_S(y|x) = P_T(y|x)$), but different marginal distributions.
- The assumption of having the same predictive distribution remains restrictive.
- When the predictive distribution is changed, negative transfer may occur.
Sample Selection Bias

- Generally, Source and Target domains also have **different distributions** mainly $P(y|x)$ or $P(y)$
  - **Class label inconsistency** among different domains
  - **Imbalanced class ratio** between source and target domains
- Re-sampling?
- But *without target labeled data*, how to do re-sampling?
Outline

1. Review of Domain Adaptation (DA)
2. Domain Selection and Multi-Source Learning
3. Optimizing Performance Measures for DA
4. DA in Heterogeneous Feature Domains
5. Conclusion
Multi-Source Learning

- **Which source domains are useful** to carry out Domain Adaptation for the target task?
  - Target domain has no or very few labeled data
  - Source domains might have different class distribution from the target one

- A natural question is how to choose **relevant samples/features/models/classifiers** from multiple source domains that fit for the target task

- Resort to Data/Domain Selection
Adaptive SVM [Yang et al, 2007]

Auxiliary Domains

\[ D^1 = (x^1_i, y^1_i) \mid_{i=1}^{n_1} \]

\[ D^2 = (x^2_i, y^2_i) \mid_{i=1}^{n_2} \]

\[ \vdots \]

\[ D^p = (x^p_i, y^p_i) \mid_{i=1}^{n_p} \]

Pre-learned Classifiers

\[ f^1(x) = \sum_{i=1}^{n_1} \alpha_i^1 y_i^1 k(x^1_i, x) \]

\[ f^2(x) = \sum_{i=1}^{n_2} \alpha_i^2 y_i^2 k(x^2_i, x) \]

\[ \vdots \]

\[ f^p(x) = \sum_{i=1}^{n_p} \alpha_i^p y_i^p k(x^p_i, x) \]

\[ \gamma_1 \]

\[ \gamma_2 \]

\[ \gamma_p \]
Adaptive SVM [Yang et al, 2007]

\[ D^1 = (x^1_i, y^1_i) \ |_{i=1}^{n_1} \]
\[ f^1(x) = \sum_{i=1}^{n_1} \alpha^1_i y^1_i k(x^1_i, x) \]
\[ \gamma_1 \]

\[ D^2 = (x^2_i, y^2_i) \ |_{i=1}^{n_2} \]
\[ f^2(x) = \sum_{i=1}^{n_2} \alpha^2_i y^2_i k(x^2_i, x) \]
\[ \gamma_2 \]

\[ \vdots \]

\[ D^p = (x^p_i, y^p_i) \ |_{i=1}^{n_p} \]
\[ f^p(x) = \sum_{i=1}^{n_p} \alpha^p_i y^p_i k(x^p_i, x) \]
\[ \gamma_p \]

Target Domain Labeled Data
\[ D^T_1 = (x^T_i, y^T_i) \ |_{i=1}^{n_1} \]
\[ \Delta f(x) = \sum_{i=1}^{n_1} \alpha^T_i y^T_i k(x^T_i, x) \]

Perturbation Function

Decision Function
\[ f^T(x) = \sum_{s} \gamma_s f^s(x) + \Delta f(x) \]

Drawbacks
- Auxiliary classifiers are fused in the target decision function, so the prediction is slow
- The importance of auxiliary classifiers is predefined.
Domain Adaptation SVM (DASVM) [L. Bruzzone et al. TPAMI ’10]

- Using target unlabeled data
  1. Using all remaining source and target labeled data to train Progressive Transductive SVM (PTSVM) and then infer the label of target unlabeled patterns
  2. By removing all source labeled patterns progressively
  3. Until only the target patterns left in the training set

- Circular validation strategy is employed to avoid poor label inference
- DASVM does **NOT** consider the predictive distributions of different domains
- DASVM can deal with a single source domain only
Predictive Distribution Matching (PDM) [Seah et al. ECML 2010, TSMC 2013]

- Two patterns having similar predictive distributions
  - Share similar predictive output
  - Measure the closeness of predictive distribution

\[ W_{i,j}^{rc} = \sum_{z=1}^{v} P^r(y_z|x_i^r)P^c(y_z|x_j^c)I[y_i = y_j]D[r \neq c]S(x_i^r, x_j^c), \]

Similarity of predictive distributions (indicates positive transfer)

- Predictive Distribution Matching (PDM) regularizer

\[ \Omega(f) = \frac{1}{n^2} \sum_{r,c=1}^{m} \sum_{i=1}^{n_r} \sum_{j=1}^{n_c} (f(x_i^r) - f(x_j^c))^2 W_{i,j}^{rc}, \]

- PDM-SVM
  - SVM + PDM regularizer
  - It can be solved by LapSVM [Belkin et al, JMLR’06]
How to Estimate Predictive Distribution?

• Many labeled data in source domains
  • Estimating source predictive distribution $P_S(y|x)$ is simple

• Few or even no labeled data in target domain
  • $P_T(y|x)$ cannot be well-estimated with limited target labeled data
  • Solution: Progressively labels certain number of target unlabeled data with pseudo-labels

• Target label inference process for target unlabeled samples is still expensive, exponential in size
Outline

1. Review of Domain Adaptation (DA)
2. Domain Selection and Multi-Source Learning
   • Healing Source Bias via Source Classifier Selection
   • DA via Distance Metric Selection
3. Optimizing Performance Measures for DA
4. DA in Heterogeneous Feature Domains
5. Conclusion
Source Classifiers

- Multiple sources $\Rightarrow$ Many source classifiers
  - Each source classifier can be trained by different algorithms (such as SVM, LR, DA methods) to speed up the label inference process of target unlabeled data
  - A source classifier is also biased towards source data

- How to choose a classifier for the target domain?
Positive and Negative Transfer

- Responses that lead to positive outcomes should be retained. Responses that lead to negative outcomes should be removed.
  - Positive outcomes refer to **Positive Transfer**
  - Negative outcomes refer to **Negative Transfer**

- How to define **Positive and Negative Transfer** without target labels?
Cluster Assumption on Target Domain

- Inspired by Maximum Margin Clustering
  - Separate the target unlabeled data into different clusters with a large margin
- A poor source classifier does not separate well target unlabeled data
- Cluster separation on target unlabeled data can be served as a criterion to measure the “goodness” of a classifier
Maximum Margin Target Label Learning (MMTLL) [Seah et al. IEEE ICDM’11]

- Select a “good” source classifier that has minimal bias towards source domains
  - A small margin separation refers to **Negative Transfer**
  - A large margin separation refers to **Positive Transfer**
  - A source classifier with a **larger margin separation** of the target unlabeled data is more preferred for the target domain

- Unlike TSVM, **NO available target labeled data** are needed in MMTLL
Algorithm

Figure 1. Maximal Margin Target Label Learning Framework
Generate Target Label Candidates

- Problem: The bias of source classifier steers towards the source distribution.

- Solution for correcting such bias:
  - The bias is computed based on the target unlabeled data.
  - A target label vector $y_s$ of the $u$ target unlabeled data is generated:
    - for each source classifier
    - and the label vector $y_s$ satisfies the balance constraint $\beta \leq \frac{1'(y_s+1)}{2} \leq u - \beta$ where $\beta$ is the parameter to control the class balance.
2D space and Linear Classifiers
2D space and Linear Classifiers
2D space and Linear Classifiers
2D space and Linear Classifiers
Target Label Space

- The number of target label candidates is at most: $Z = (u - 2\beta) \sum_{i=1}^{m} \frac{m!}{i!(m-i)!}$
- The target label space becomes $M = \{\hat{y} = \sum_{s=1}^{Z} g_s y_s, \sum_{s=1}^{Z} g_s = 1, g_s \geq 0\}$
- Next step is to learn the weight of each target label vector $y_s$
A label vector of 1st classifier

\[ g_1 = 0.3 \]
A label vector of 2nd classifier

\[ g_2 = 0.3 \]
A label vector of 3rd classifier

\[ g_3 = 0.4 \]
Learning Target Classifier using Maximum Margin Margin Criterion

- MMTLL maximizes the margin separation on the target unlabeled data:

Primal of SVM

\[
\min_{\hat{y} \in M} \min_{\mathbf{w}, \rho, \xi_i} \frac{1}{2} \| \mathbf{w} \|^2 - \rho + C \sum_{i=1}^{u} \xi_i
\]

s.t. \( \hat{y}_i \mathbf{w}' \varphi(\mathbf{x}_i) \geq \rho - \xi_i, \forall i = 1, \ldots, u \)

- Equivalent to choosing a linear combination of target label vectors that minimize the empirical risk
Final Formulation of MMTLL

This can be solved by Multiple Kernel Learning methods:

\[
\min_{d \in D} \left\{ \max_{\alpha \in A} -\frac{1}{2} \alpha' \left( \sum_{t:y_t \in M} d_t K \otimes y_t y_t' \right) \alpha \right\}
\]

where

\[
D = \{ d \mid \sum_{t:y_t \in M} d_t = 1, d_t \geq 0, \ \forall t : y_t \in M \},
\]

\[
A = \{ \alpha \mid \sum_{i=1}^{u} \alpha_i = 1, 0 \leq \alpha_i \leq C, \ i = 1, \ldots, u \}.
\]
\[ y_t = \arg\max_{y \in M_2} \frac{1}{2} \alpha'(K \otimes yy') \alpha, \text{ where } M_2 = \{y_1, \ldots, y_Z\} \]

1) Select most violated constraint

Add \( y_t \)

2) Multiple Kernel Learning

\( d_t \) as the weight of \( y_t \)

Converge

\text{NO}

\text{YES} \quad \hat{y} = \sum_{t: y_t \in M_2} d_t y_t
Experimental setup

- **Datasets:**
  1. Sentiment (book, DVDs, Electronics, Kitchen Appliances)
  2. Newsgroups

<table>
<thead>
<tr>
<th>Domain</th>
<th>Category comp</th>
<th>Category rec</th>
<th>Category sci</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source 1</td>
<td>windows.x</td>
<td>motorcycles</td>
<td>electronics</td>
</tr>
<tr>
<td>Source 2</td>
<td>sys.ibm.pc.hardware</td>
<td>sport.baseball</td>
<td>med</td>
</tr>
<tr>
<td>Source 3</td>
<td>sys.mac.hardware</td>
<td>sport.hockey</td>
<td>space</td>
</tr>
<tr>
<td>Target</td>
<td>graphics</td>
<td>autos</td>
<td>crypt</td>
</tr>
</tbody>
</table>

- **Performance metric**
  - **Balanced Accuracy:** \[0.5 \left( \frac{tp}{tp+fn} + \frac{tn}{tn+fp} \right)\]
  - Since domains have different distributions:
    - Target Positive Class Ratio (**TPCR**): number of positive samples in the **target** domain
    - Source Positive Class Ratio (**SPCR**): number of positive samples in the **source** domains.
Baseline methods

- $I_S$-SVM$_{\text{Best}}$: SVM for each source domain
- $2S$-SVM$_{\text{Best}}$: SVM for each unique pair of source domains
- $MCC = 3S$-SVM: Multiple Convex Combination
- $LG$-$MMC$: Label Generation Maximum Margin Clustering
- $KMM_{\text{Best}}$: Kernel Mean Matching using weighted SVM for each source domain
- $TCA_{\text{Best}}$: Transfer Component Analysis using SVM on the transfer components
Sentiment Results

(a) DVDs TPCR=0.3
(b) Electronics TPCR=0.3
(c) Kitchen Appliances TPCR=0.3
(d) DVDs TPCR=0.5
(e) Electronics TPCR=0.5
(f) Kitchen Appliances TPCR=0.5
Newsgroup Results

(a) comp vs. rec TPRC=0.3
(b) comp vs. sci TPRC=0.3
(c) rec vs. sci TPRC=0.3
(d) comp vs. rec TPRC=0.5
(e) comp vs. sci TPRC=0.5
(f) rec vs. sci TPRC=0.5
Automatic Source Classifier Selection

Target DVDs TPCR=0.3

Rec vs. Sci TPCR=0.3
Computational Performance

Kitchen appliances as target domain at TPCR=0.3

- MMTLL scales much faster than DASVM
Summary of MMTLL

- Existing DA methods
  - cannot automatically choose relevant source domains
  - are very sensitive to the unknown target class ratio

- MMTLL can integrate any DA methods
  - The optimal combination of source classifiers for the target domain can be learned by convex optimization
  - The bias and the weight of the source classifiers are learned based on the target unlabeled data only

- Unlike previous DA methods, this framework
  - does NOT assume shared support and having similar predictive distribution between source and target domains
  - is robust to diverse target class ratios
Outline

1. Review of Domain Adaptation (DA)
2. Domain Selection and Multi-Source Learning
   - Healing Source Bias via Source Classifier Selection
   - DA via Distance Metric Selection
3. Optimizing Performance Measures for DA
4. DA in Heterogeneous Feature Domains
5. Conclusion
DA via Distance Metric Selection

- DAs focus on learning Classification/Regression models
- Can DAs be applied to Clustering?
  - Learning/Selecting distance metric for clustering in DA
Solving NP-Hard Optimizations from The Past Experiences [Feng et al, CEC’12, Best Student Paper Nomination]

- Can DAs be applied to general optimizations?
  - Optimization methods generally start a search from scratch or ground zero state
  - Problems seldom exist in isolation
  - Useful information from past problems may be useful for solving future problems

- Leveraging models learned from the past via **distance metric selection** to learn a new model in the target task
Capacitated Arc Routing Problem (CARP)

These consist of problems of diverse properties in terms of vertices size, graph topologies, etc.

- Servicing a set of streets
- Using a fleet of capacity constrained vehicles
- These vehicles start and end at the central depot
- Minimize the total routing cost involved

NP-Hard Combinatorial optimization problems
An Example of CARP

How to reuse the optimized solution from previous CARPs?

CARPs can be initialized by clustering + TSPs
Learning Distance Metric from Past CARPs

- CARP can be reduced to a set of individual Travelling Salesman Problems (TSPs) for each vehicle

\[ X = \begin{bmatrix} V_1 & V_2 & \ldots & V_{14} \\ V_{11} & V_{21} & \ldots & V_{141} \\ V_{12} & V_{22} & \ldots & V_{142} \\ \vdots & \vdots & \ddots & \vdots \\ V_{1p} & V_{2p} & \ldots & V_{14p} \end{bmatrix} \] defines the vertices

\[ Y = \begin{bmatrix} 1 & 1 & 1 & \ldots & -1 & -1 \\ 1 & 1 & 1 & \ldots & -1 & -1 \\ 1 & 1 & 1 & \ldots & -1 & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -1 & -1 & -1 & \ldots & 1 & 1 \\ -1 & -1 & -1 & \ldots & 1 & 1 \end{bmatrix} \] defines the grouping of vertices in different TSPs

- Given the optimal grouping \( Y \) of a previous CARP, one can learn a metric \( M \) for vertices by maximizing the dependence with \( Y \):

\[
\max_{M} \text{tr} (HKHY) \\
\text{Hilbert-Schmidt Independence Criterion}
\]

\[
s.t. \ K = X'MX, \ M \succeq 0, \ H = I - \frac{1}{n} \mathbf{1}\mathbf{1}'
\]
A Pool of Previously Learned Distance Metrics

- Given \( \{M_1, M_2, \ldots, M_n\} \)

- The target metric is \( M_t = \sum_{i=1}^{n} \mu_i M_i \) subject to \( \mathcal{N} = \{\mu| \sum_{i=1}^{n} \mu_i = 1, \mu_i \in \{0,1\}, \forall i = 1 \ldots n\} \)
1. For a fixed $Y$, we learn the target distance metric:

$$M_t = \sum_{i=1}^{n} \mu_i M_i$$

where $\mu$ is learned by maximizing dependence and similarity $S_i$ between source and target domains:

$$\text{HSIC} \quad \text{Similarity between domains}$$

$$\max_{\mu \in \mathcal{N}} \text{tr}(HX'M_tXHY) + \sum_{i=1}^{n} \mu_i S_i$$

2. For a fixed $\mu$, we learn the clustering $Y$ by dependence maximization

3. Repeat the above steps until convergence

Based on $Y$, CARP can be reduced to solving a set of TSPs
Empirical Studies of CARPs in DA

- Optimization Solver: ILMA [Mei et al. (2009)]
- ILMA with Variants of Population Initialization Procedures
  - ILMA
  - ILMA-R
  - ILMA-K
  - ILMA-L (DA method)
A pool of metrics are learned from past experiences of CARP datasets in egl, namely “E1A”, “E1B”, “E2A”, “E3A”, “E4A”, “S1A”, “S1B”, “S2A”, “S3A”, “S4A” by using the results reported in [Mei et al. 2009].

Detailed properties of the unseen or unsolved CARP benchmarks:

<table>
<thead>
<tr>
<th></th>
<th>“E” Series</th>
<th>“S” Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>E1C</td>
<td>E2B</td>
</tr>
<tr>
<td></td>
<td>E2C</td>
<td>E3B</td>
</tr>
<tr>
<td>#Vertices</td>
<td>77</td>
<td>77</td>
</tr>
<tr>
<td>#Edges required to be served</td>
<td>51</td>
<td>72</td>
</tr>
<tr>
<td>#Edges</td>
<td>98</td>
<td>98</td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E2B</td>
<td>6317</td>
<td>6337.2</td>
<td>6317</td>
<td>6343.0</td>
<td>6317</td>
<td>6337.6</td>
<td>6317</td>
<td>6333.6</td>
</tr>
<tr>
<td>E2C</td>
<td>8335</td>
<td>8350.6</td>
<td>8335</td>
<td>8350.1</td>
<td>8335</td>
<td>8351.2</td>
<td>8335</td>
<td>6347.2</td>
</tr>
<tr>
<td>E3B</td>
<td>7777</td>
<td>7799.4</td>
<td>7777</td>
<td>7793.5</td>
<td>7775</td>
<td>7790.9</td>
<td>7775</td>
<td>7788.5</td>
</tr>
<tr>
<td>E4B</td>
<td>8998</td>
<td>9057.4</td>
<td>8990</td>
<td>9062.5</td>
<td>8998</td>
<td>9058.1</td>
<td>8988</td>
<td>9050.0</td>
</tr>
<tr>
<td>E4C</td>
<td>11609</td>
<td>11714.6</td>
<td>11594</td>
<td>11728.0</td>
<td>11606</td>
<td>11721.2</td>
<td>11542</td>
<td>16648.1</td>
</tr>
<tr>
<td>S1C</td>
<td>8519</td>
<td>8583.9</td>
<td>8518</td>
<td>8583.5</td>
<td>8518</td>
<td>8582.2</td>
<td>8518</td>
<td>8563.7</td>
</tr>
<tr>
<td>S2C</td>
<td>16490</td>
<td>16601.8</td>
<td>16477</td>
<td>16617.7</td>
<td>16482</td>
<td>16620.8</td>
<td>16456</td>
<td>16594.3</td>
</tr>
<tr>
<td>S3C</td>
<td>17285</td>
<td>17393.3</td>
<td>17282</td>
<td>17396.4</td>
<td>17291</td>
<td>17362.3</td>
<td>17253</td>
<td>17353.5</td>
</tr>
<tr>
<td>S4B</td>
<td>16394</td>
<td>16529.6</td>
<td>16435</td>
<td>16541.8</td>
<td>16379</td>
<td>16518.9</td>
<td>16358</td>
<td>16506.1</td>
</tr>
<tr>
<td>S4C</td>
<td>20608</td>
<td>20804.2</td>
<td>20653</td>
<td>20810.5</td>
<td>20625</td>
<td>20803.5</td>
<td>20581</td>
<td>20776.4</td>
</tr>
</tbody>
</table>

* Superior performance are highlighted in bold font
Summary

- The learned metric for previous clustering results can be used for learning the target metric of a new clustering problem
- CARP can be reduced to clustering + TSPs
- The optimization cost of a new CARP can be vastly reduced
- DA for CARP can achieve better performance than non DA methods
Outline

1. Review of Domain Adaptation (DA)
2. Domain Selection and Multi-Source Learning
3. Optimizing Performance Measures for DA
4. DA in Heterogeneous Feature Domains
5. Conclusion
Coreference Resolution (CR) in NLP

Coreference resolution is the process of determining whether two mentions (expressions) in natural language refer to the same entity.

Eg. [Barack Obama] \(_1^1\) nominated [Hillary Rodham Clinton] \(_2^2\) as [[his] \(_3^1\) secretary of state] \(_4^3\) on [Monday] \(_5^4\). [He] \(_6^1\) …

- Finding mention pairs to form entities is usually formulated as *supervised clustering*.
- Most CR methods focus on the within-domain case (i.e., training and test data come from the same distribution).
Coreference Resolution

Labeled Mention Pairs

Obama ↔ US President
Hillary ↔ Secretary of State

Feature Extractions

Grammatical features
Semantic features
Lexical, Positional features

Prediction

Barack Obama ↔ US President
Hillary ↔ Secretary of State
Hillary Clinton ↔ US President

Cluster Model Building

Learning methods: Mention-pair models, Mention-ranking models, Clustering ranking models, etc.
CR on Resource-Poor Domains

- Mention pairs that refer to the same entity require linguistic experts to annotate
  \[ \Rightarrow \text{Labeled mention pairs are difficult to obtain} \]
- In some resource poor domains (e.g., medical reports) where only a few labeled mention pairs are available, or even no available expert
  \[ \Rightarrow \text{It is hard to learn a robust CR model} \]

Can we leverage the labeled mention pairs from other resource-rich domains to the target domain?
Performance Measures for CR

- The number of clusters (entities) is **unknown**
- The cluster size is also unknown and **diverse**
  - Some entities may contain hundreds of mentions
  - Some entities contain only one single mention
- To measure the quality of clustering with diverse characteristics, many performance measures are required to report
  - MUC F-measure,
  - B-CUBE F-measure,
  - CEAF F-measure,
  - BLANC F-measure, etc.
Optimizing Performance Measures for DA

- The design and generalization analysis of existing DA methods focus on accuracy only.
- How to optimize other performance measures in DA?
- The commonly used performance measures in CR are neither decomposable nor enumerable.
  \[ \Rightarrow \text{SVMperf CANNOT be applied} \]
- Many clustering models learned from different source domains are available.

Can we leverage these models for the target task?
Adaptive Ensemble Approach for DA
[Yang et al. EMNLP’12]
Ensemble Creation

- Each base model is constructed on its own domain
  - Train an SVM with a tradeoff parameter $C$ to classify each mention-pair to be coreferent or not
  - Employ closest-first or best-first clustering methods to construct the entities with a threshold $\tau$
- By varying $C$ and $\tau$, we create a domain-specific ensemble $\mathcal{F}^v = \{f^1, \ldots, f^l\}$
- For multiple domains, we gather all the domain-specific ensembles into a grand ensemble $\mathcal{F} = \mathcal{F}^{s_1} \cup \ldots \cup \mathcal{F}^{s_p} \cup \mathcal{F}^t$
Cross Domain Distance Learning

To learn a distance measure between documents across domains that is specific to the user-specific objective measure $\Lambda(\cdot,\cdot)$:

- Feature representation of a document: $\Phi(D_i) = \sum_{(a,b) \in \varepsilon_i} \varphi_{a,b}$
- Parameterized distance: $\text{Dist} \left( D_i^{S_u}, D_j^{(t)}; \mu \right) = \mu' \Delta(D_i^{S_u}, D_j^{(t)})$ where $\Delta(D_i^{S_u}, D_j^{(t)}) = (\Phi(D_i^{S_u}) - \Phi(D_i^{t})) \odot (\Phi(D_i^{S_u}) - \Phi(D_i^{t}))$

Learn $\mu$ by solving

$$\min_{\mu} \sum_{j=1}^{N^{(t)}} \sum_{D_i^{S_u} \in I(D_i^{t}; S_u)} \text{Dist} \left( D_i^{S_u}, D_j^{(t)}; \mu \right): \mu' 1 \geq B$$

$I(D_i^{t}; S_u)$ is the set of documents in the source domain $S_u$ that are similar to $D_i^{t}$ in the sense that they have the best performing model under the measure $\Lambda(\cdot,\cdot)$.
Decision Inference

For each target test document $D_j^{(t)}$, we predict it using the best performing based model as

$$f_j^{(t)} = \arg\max_{D_p \in \mathcal{N}(D_j^{(t)}), f \in \mathcal{F}} \Lambda(D_p; f(D_p))$$

- $\mathcal{N}(D_j^{(t)})$ is the set of $k$ nearest neighbour documents for $D_j^{(t)}$ from the all labeled source documents based on the learned distance measure.
Copora

- **MUC-6**
- **ACE2005 (6 domains)**
  - Broadcast Conversation (BC);
  - Broadcast News (BN);
  - Newswire (NW);
  - Usenet (UN);
  - WebBlogs (WL);
  - Conversation al Telephone Speech (CTS)

<table>
<thead>
<tr>
<th>Domain</th>
<th>$N^{(t)}$</th>
<th>$M^{(t)}$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC-6</td>
<td>30</td>
<td>30</td>
<td>60</td>
</tr>
<tr>
<td>BC</td>
<td>48</td>
<td>12</td>
<td>60</td>
</tr>
<tr>
<td>BN</td>
<td>181</td>
<td>45</td>
<td>226</td>
</tr>
<tr>
<td>CTS</td>
<td>31</td>
<td>8</td>
<td>39</td>
</tr>
<tr>
<td>NW</td>
<td>85</td>
<td>21</td>
<td>106</td>
</tr>
<tr>
<td>UN</td>
<td>39</td>
<td>10</td>
<td>49</td>
</tr>
<tr>
<td>WL</td>
<td>95</td>
<td>24</td>
<td>119</td>
</tr>
</tbody>
</table>

- $N^{(t)}$ and $M^{(t)}$ are the training and test set sizes
- 5 labeled documents are used in domain adaptation setting
## Domain Adaptation Setting in B-Cube

<table>
<thead>
<tr>
<th></th>
<th>Within-domain</th>
<th>Grand ensemble</th>
<th>Domain-adaptation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$S_c$</td>
<td>$S_b$</td>
<td>$k=1$</td>
</tr>
<tr>
<td>BC</td>
<td>58.0</td>
<td>65.1</td>
<td>65.0</td>
</tr>
<tr>
<td>BN</td>
<td>72.7</td>
<td>73.8</td>
<td>75.0</td>
</tr>
<tr>
<td>CTS</td>
<td>63.2</td>
<td>62.1</td>
<td>65.7</td>
</tr>
<tr>
<td>NW</td>
<td>54.9</td>
<td>54.6</td>
<td>73.6</td>
</tr>
<tr>
<td>UN</td>
<td>66.5</td>
<td>42.7</td>
<td>67.2</td>
</tr>
<tr>
<td>WL</td>
<td>68.6</td>
<td>73.2</td>
<td>73.0</td>
</tr>
<tr>
<td>Average</td>
<td>64.0</td>
<td>61.9</td>
<td>69.9</td>
</tr>
</tbody>
</table>

- B-Cube F-measure is reported
- Our DA method outperform other baselines
## Domain Adaptation Setting in CEAF

<table>
<thead>
<tr>
<th></th>
<th>Within-domain</th>
<th>Grand ensemble</th>
<th>Domain-adaptation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$S_c$</td>
<td>$S_b$</td>
<td>$k=1$</td>
</tr>
<tr>
<td>BC</td>
<td>55.7</td>
<td>43.7</td>
<td>56.9</td>
</tr>
<tr>
<td>BN</td>
<td>65.8</td>
<td>67.2</td>
<td>65.9</td>
</tr>
<tr>
<td>CTS</td>
<td>56.0</td>
<td>51.0</td>
<td>56.6</td>
</tr>
<tr>
<td>NW</td>
<td>52.7</td>
<td>55.0</td>
<td>66.4</td>
</tr>
<tr>
<td>UN</td>
<td>64.0</td>
<td>39.1</td>
<td>63.6</td>
</tr>
<tr>
<td>WL</td>
<td>70.3</td>
<td>64.2</td>
<td>68.1</td>
</tr>
<tr>
<td>Average</td>
<td>60.7</td>
<td>53.4</td>
<td>62.9</td>
</tr>
</tbody>
</table>

- CEAF F-measure is reported
- Our DA method outperforms other baselines
Within Domain Setting

- Our method can optimize performance measure for coreference resolution
- When $k = 1$, our method achieves the best performance
Summary of Cross Domain Distance Learning

- Learning a distance that finds a source document having the best predicting model for the target document
- Automatically choosing the best model for each individual target document rather than a fixed model
- The first DA work to optimize any user-specific performance measures
Outline

1. Review of Domain Adaptation (DA)
2. Domain Selection and Multi-Source Learning
3. Optimizing Performance Measures for DA
4. DA in Heterogeneous Feature Domains
5. Conclusion
Heterogeneous Domain Adaptation

- Source and target data have different feature dim.
  - English vs. Chinese
  - SIFT features vs. SURF features for Images
  - Text vs. Images
Sentiments in Different Languages

- Many existing reviews with rating on *Happy Feet Two* are written in English
- When *Happy Feet Two* is broadcast in China, how’s the rating of Chinese reviews?
- Source and Target data are represented in different feature spaces
- Can we still apply the model learned in English to Chinese reviews?
Existing Heterogeneous DA methods

**HeMap** [Shi et al., ICDM’10]
- It finds the projection matrices for a common feature subspace as well as learns the optimal projected data from both domains
- *Unsupervised*

**DAMA** [Wang et al., IJCAI’11]
- It learns a common feature subspace by utilizing the class labels of the source and target labeled data and manifold alignment

**ARC-t** [Kulis et al., CVPR’11]
- Distance metric learning using sign information
- It learns an asymmetric transformation metric between the different feature spaces for all classes
ARC-t [Kulis et al., CVPR’11]

1. Multi-Domain Data

2. Generate Constraints, Learn W

3. Map via W

Figures are adapted from Saenko
Outline

1. Review of Domain Adaptation (DA)
2. Domain Selection and Multi-Source Learning
3. Optimizing Performance Measures for DA
4. DA in Heterogeneous Feature Domains
   ◦ Heterogeneous Feature Augmentation
5. Conclusion
Heterogeneous Feature Augmentation [Duan et al. ICML’12]

- Mapping heterogeneous features from two domains into a common subspace
  - $\mathbf{x}^s \in \mathbb{R}^{d_s}$ and $\mathbf{x}^t \in \mathbb{R}^{d_t}$, $d_s \neq d_t$
  - Projection $\mathbf{P} \in \mathbb{R}^{d_c \times d_s}$: $\mathbf{x}^s \rightarrow \mathbf{P} \cdot \mathbf{x}^s$
  - Projection $\mathbf{Q} \in \mathbb{R}^{d_c \times d_t}$: $\mathbf{x}^t \rightarrow \mathbf{Q} \cdot \mathbf{x}^t$

![Diagram showing mapping of heterogeneous features into a common subspace]
Feature Augmentation for the same feature space:

\[ \varphi_s(x_s) = \begin{bmatrix} x_s^s \\ x_s \\ 0 \end{bmatrix} \quad \text{and} \quad \varphi_t(x_t) = \begin{bmatrix} x_t^t \\ 0 \\ x_t \end{bmatrix} \]

- New feature mappings \( \varphi_s(x_s) \) and \( \varphi_t(x_t) \) for source and target data, respectively
- Similarities between data from the same domain is doubled by incorporating the original features [Daumé III, ACL’07]

\[ \tilde{K}(x_i, x_j) = \begin{cases} 2 \cdot K(x_i, x_j), & \text{if } x_i \text{ and } x_j \text{ are from the same domain} \\ K(x_i, x_j), & \text{if } x_i \text{ and } x_j \text{ are from different domains} \end{cases} \]
Heterogeneous Feature Augmentation (HFA)

\[ \varphi_s(x^s) = \begin{bmatrix} Px^s \\ x^s \\ 0_{d_t} \end{bmatrix} \quad \text{and} \quad \varphi_t(x^t) = \begin{bmatrix} Qx^t \\ 0_{d_s} \\ x^t \end{bmatrix} \]

- New feature mappings \( \varphi_s(x^s) \) and \( \varphi_t(x^t) \) for source and target data in heterogeneous feature domains, respectively.
- Now, data with heterogeneous features can be compared in the common subspace.
- Similarities between data from the same domain can also be enhanced by incorporating the original features.
Primal Formulation

- HFA Formulation with SVM

\[
\min_{P,Q,w,b,\xi_i^s,\xi_i^t} \frac{1}{2} \|w\|^2 + C \left( \sum_{i=1}^{n_s} \xi_i^s + \sum_{i=1}^{n_t} \xi_i^t \right)
\]

s.t. \[y_i^s (w' \varphi_s(x_i^s) + b) \geq 1 - \xi_i^s, \quad \xi_i^s \geq 0;\]
\[y_i^t (w' \varphi_t(x_i^t) + b) \geq 1 - \xi_i^t, \quad \xi_i^t \geq 0;\]
\[\|P\|_F^2 \leq \lambda_p, \|Q\|_F^2 \leq \lambda_q\]

- \[w = [w_c', w_s', w_t']': A \text{ feature weight vector}\]
- \[\xi_i^s \text{ and } \xi_i^t: \text{ Slack variables}\]
- \[\lambda_p \text{ and } \lambda_q: \text{ Pre-defined parameters}\]
Dual Formulation

- The Dual w.r.t. \( \{\mathbf{w}, b, \xi_i^s, \xi_i^t\} \):

\[
\begin{align*}
\min_{P,Q} \max_{\alpha} & \quad \mathbf{1}_{n_s+n_t}' \alpha - \frac{1}{2} (\alpha \circ \mathbf{y})' K_{P,Q} (\alpha \circ \mathbf{y}) \\
\text{s. t.} & \quad \mathbf{y}' \alpha = 0, \mathbf{0}_{n_s+n_t} \leq \alpha \leq C \mathbf{1}_{n_s+n_t}; \\
& \quad \|P\|_F^2 \leq \lambda_p, \|Q\|_F^2 \leq \lambda_q
\end{align*}
\]

- \( \mathbf{\alpha} \) : a vector of dual variables; \( \mathbf{y} \) : a vector of training labels

- \( K_{P,Q} = \begin{bmatrix}
X'_s (I_{n_s} + \mathbf{P}'\mathbf{P}) X_s & X'_s \mathbf{P}' \mathbf{Q} X_t \\
X'_t \mathbf{Q}'\mathbf{P} X_s & X'_t (I_{n_t} + \mathbf{Q}'\mathbf{Q}) X_t
\end{bmatrix} \)

- It is nontrivial to determine the optimal dimension \( d_c \) for the common space
Simplification of $K_{P, Q}$

By introducing a transformation metric

$$H = [P, Q]' [P, Q] \in \mathbb{R}^{(d_s+d_t) \times (d_s+d_t)}$$

and

$$L_s = \begin{bmatrix} I_{d_s} \\ 0_{d_t \times d_s} \end{bmatrix} X_s, \quad L_t = \begin{bmatrix} 0_{d_s \times d_t} \\ I_{d_s} \end{bmatrix} X_t, \quad \lambda = \lambda_p + \lambda_q$$

then we have

$$K_H = \begin{bmatrix} X'X_s + L'_s H L_s & L'_s H L_t \\ L'_t H L_s & X_t'X_t + L'_t H L_t \end{bmatrix}$$

$\Rightarrow$ get rid of $d_c$
Optimization

- Solution: Iteratively update $H$ using SDP and the dual variable $\alpha$ using SVM

- Linear HFA
  - $H$ is linear, so it may not be effective for nonlinear tasks
  - Infeasible for applications with very high dimensional data

- Can we learn $H$ with its size dependent on the number of training samples?
  - Kernelized HFA
  - Nonlinear transformation metric
    \[
    \tilde{H} = [\tilde{P}, \tilde{Q}'] [\tilde{P}, \tilde{Q}] \in \mathbb{R}^{(n_s+n_t) \times (n_s+n_t)}
    \]
Multi-Domain Object Dataset [Kulis et al., CVPR’11]

- 4,106 images with 31 categories from three sources

<table>
<thead>
<tr>
<th>Domain</th>
<th># dim</th>
<th># total imgs</th>
<th># training imgs per category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>amazon</td>
<td>800</td>
<td>2,813</td>
<td>20</td>
</tr>
<tr>
<td>webcam</td>
<td>800</td>
<td>795</td>
<td>8</td>
</tr>
<tr>
<td>Target</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dslr</td>
<td>600</td>
<td>498</td>
<td>3</td>
</tr>
</tbody>
</table>

- **Source**: Amazon or webcam; **Target**: dslr
- Test data: The remaining dslr images are used
- Classification accuracy

\[
ACC = \frac{\text{# correctly classified samples}}{\text{# total test samples}}
\]
Multi-Domain Object Dataset

31 categories

... file cabinet headphones keyboard laptop letter tray ...

amazon →

dSLR →

webcam →

3 domains

Figure is taken from Saenko
Reuters Multilingual Dataset [Amini et al., NIPS’09]

- 11,547 documents with 6 classes from 5 sources

<table>
<thead>
<tr>
<th>Source</th>
<th>Domain</th>
<th># dim after PCA</th>
<th># total docs</th>
<th># training docs per class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>English</td>
<td>1,131</td>
<td>18,758</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>French</td>
<td>1,230</td>
<td>26,648</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>German</td>
<td>1,417</td>
<td>29,953</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>Italian</td>
<td>1,041</td>
<td>24,039</td>
<td>100</td>
</tr>
<tr>
<td>Target</td>
<td>Spanish</td>
<td>807</td>
<td>11,547</td>
<td>5/7/10/15/20</td>
</tr>
</tbody>
</table>

- Source: English, French, German or Italian; Target: Spanish
- Test data: The remaining Spanish documents are used
- Classification accuracy

\[
ACC = \frac{\text{# correctly classified samples}}{\text{# total test samples}}
\]
Experiments

Results

- **Object dataset** [Kulis et al., CVPR’11]

<table>
<thead>
<tr>
<th>Source Domain</th>
<th>SVM_T</th>
<th>KCCA</th>
<th>HeMap</th>
<th>DAMA</th>
<th>ARC-t</th>
<th>HFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>amazon</td>
<td>52.9 ± 3.1</td>
<td>46.3 ± 2.7 (51.0)</td>
<td>42.8 ± 2.4</td>
<td>53.3 ± 2.3</td>
<td>53.1 ± 2.4 (53.2)</td>
<td>55.4 ± 2.8</td>
</tr>
<tr>
<td>webcam</td>
<td></td>
<td>46.7 ± 2.8</td>
<td>42.2 ± 2.6</td>
<td>53.2 ± 3.2</td>
<td>53.0 ± 3.2</td>
<td>54.3 ± 3.7</td>
</tr>
</tbody>
</table>

- **Reuters multilingual dataset** [Amini et al., NIPS’09]

<table>
<thead>
<tr>
<th>Source Domain</th>
<th>SVM_T</th>
<th>KCCA</th>
<th>HeMap</th>
<th>DAMA</th>
<th>ARC-t</th>
<th>HFA</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>72.6 ± 2.3</td>
<td>71.4 ± 3.2</td>
<td>65.7 ± 3.1</td>
<td>72.4 ± 2.4</td>
<td>72.9 ± 2.0</td>
<td>75.3 ± 1.7</td>
</tr>
<tr>
<td>French</td>
<td></td>
<td>72.8 ± 2.8</td>
<td>64.2 ± 4.2</td>
<td>72.8 ± 2.0</td>
<td>73.5 ± 1.8</td>
<td>75.7 ± 1.6</td>
</tr>
<tr>
<td>German</td>
<td></td>
<td>73.8 ± 2.2</td>
<td>64.6 ± 3.6</td>
<td>72.9 ± 2.3</td>
<td>74.7 ± 1.6</td>
<td>76.1 ± 1.5</td>
</tr>
<tr>
<td>Italian</td>
<td></td>
<td>73.8 ± 2.1</td>
<td>65.8 ± 2.3</td>
<td>73.3 ± 2.1</td>
<td>74.0 ± 2.0</td>
<td>75.8 ± 1.8</td>
</tr>
</tbody>
</table>

- Our HFA method is significantly better than the other methods under both settings, judged by the t-test with a significance level at 0.05
Experiments

- **Results w.r.t. # Target Training Samples Per Class**
  - Reuters multilingual dataset [Amini et al., NIPS’09]
  - HFA emerges as superior to others especially when there are few target labeled data.
Convergence Analysis

- “back_pack” on the object dataset
- “C15” on the Reuters multilingual dataset
- Less than 80 and 40 iter. on both datasets
Outline

1. Review of Domain Adaptation (DA)
2. Domain Selection and Multi-Source Learning
3. Optimizing Performance Measures for DA
4. DA in Heterogeneous Feature Domains
5. Conclusion
Conclusion

• In this tutorial, we discuss:
  1. How to choose relevant models from source domains for the target task
  2. How to optimize the performance measure for DA
  3. How to do heterogeneous DA
Future Work

- Are there any other better criteria to detect positive and negative transfers?
- How to optimize performance measures for DA without target labeled data?
- How to do domain selection in heterogeneous domains?
- Generalization Analysis for Heterogeneous DA
References for Domain selection and Multi-source learning


- Lixin Duan, Ivor W. Tsang, Dong Xu, Tat-Seng Chua. Domain Adaptation from Multiple Sources via Auxiliary Classifiers. Proceedings of the 26th International Conference on Machine Learning (ICML 2009), Montreal, Quebec, June 2009.
References for Domain Adaptation

- John Blitzer, Ryan McDonald, and Fernando Pereira. Domain Adaptation with Structural Correspondence Learning. EMNLP 2006.
Other References
