2019 CS420, Machine Learning, Lecture 15

Transfer Learning

Weinan Zhang

Shanghai Jiao Tong University

http://wnzhang.net

http://wnzhang.net/teaching/cs420/index.html

Transfer Learning Materials

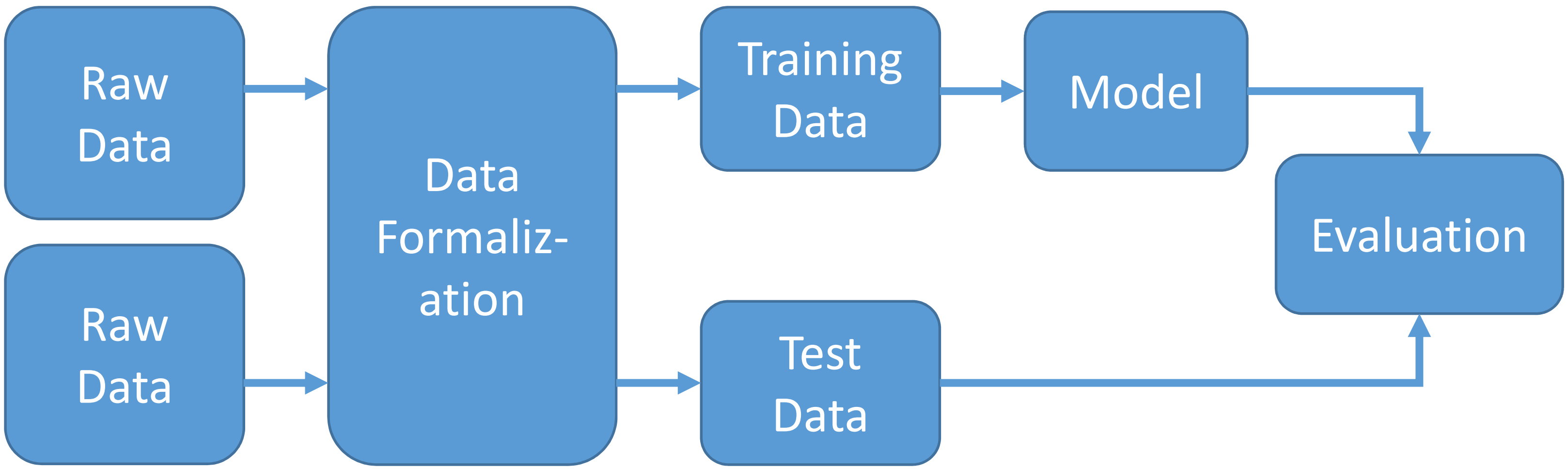
Our course on TL is mainly based on the materials from Prof. Qiang Yang and his students

Prof. Qiang Yang

- Chair Professor, Department Head of CSE, HKUST
- http://www.cs.ust.hk/~qyang/
- SJ Pan, Q Yang. A survey on transfer learning. IEEE TKDE 2010.
- 4000+ citations on this survey paper

Machine Learning Process

$$\min_{\theta} \frac{1}{N} \sum_{(x_i,y_i)\in D_{\text{train}}} \mathcal{L}(y_i, f_\theta(x_i)) + \lambda||\theta||_2^2$$



$$\text{Test Error} = \frac{1}{N} \sum_{(x_i,y_i)\in D_{\text{test}}} \mathcal{L}(y_i, f_\theta(x_i))$$

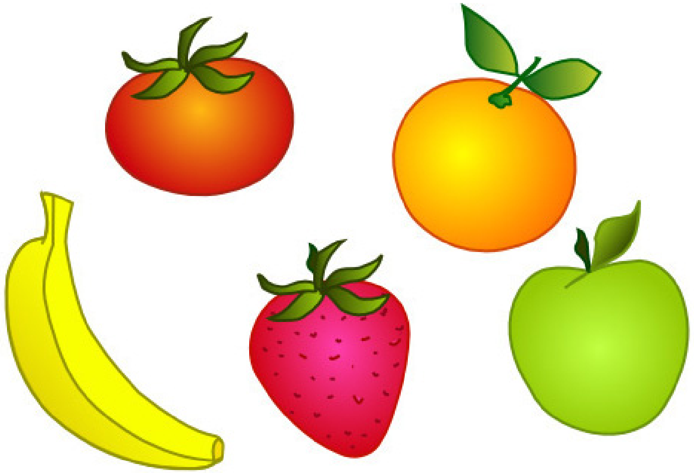
- Assumption: training and test data has the same distribution

Practical Cases

- Data distributions *p*(*x*) change across different domains or vary over time

$$\mathcal{X}_S \neq \mathcal{X}_T \quad \text{or} \quad p_S(x) \neq p_T(x)$$

Real images



Cartoon images

Practical Cases

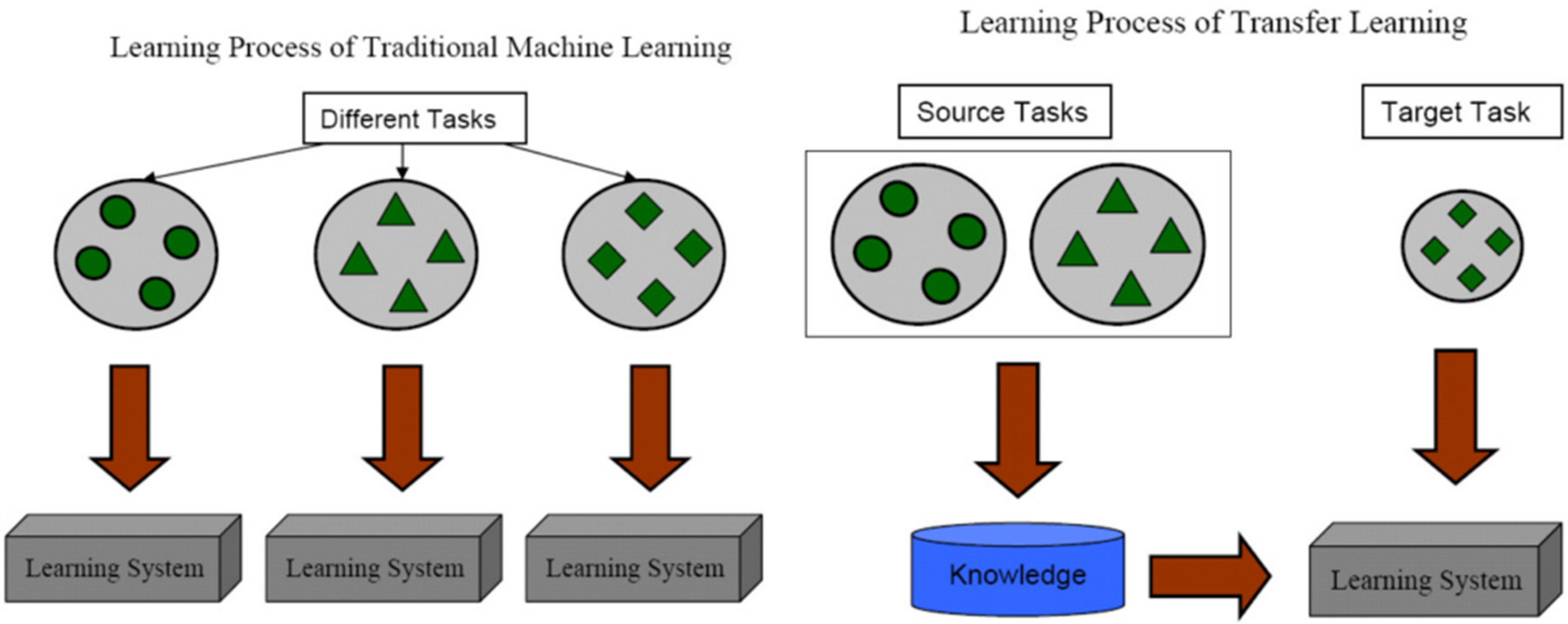
- Data dependencies $p(y|x)$ could be also different

$$\mathcal{Y}_S \neq \mathcal{Y}_T \quad \text{or} \quad p_S(y|x) \neq p_T(y|x)$$

Apple recognition

Pear recognition

Transfer Learning



(a) Traditional Machine Learning

(b) Transfer Learning

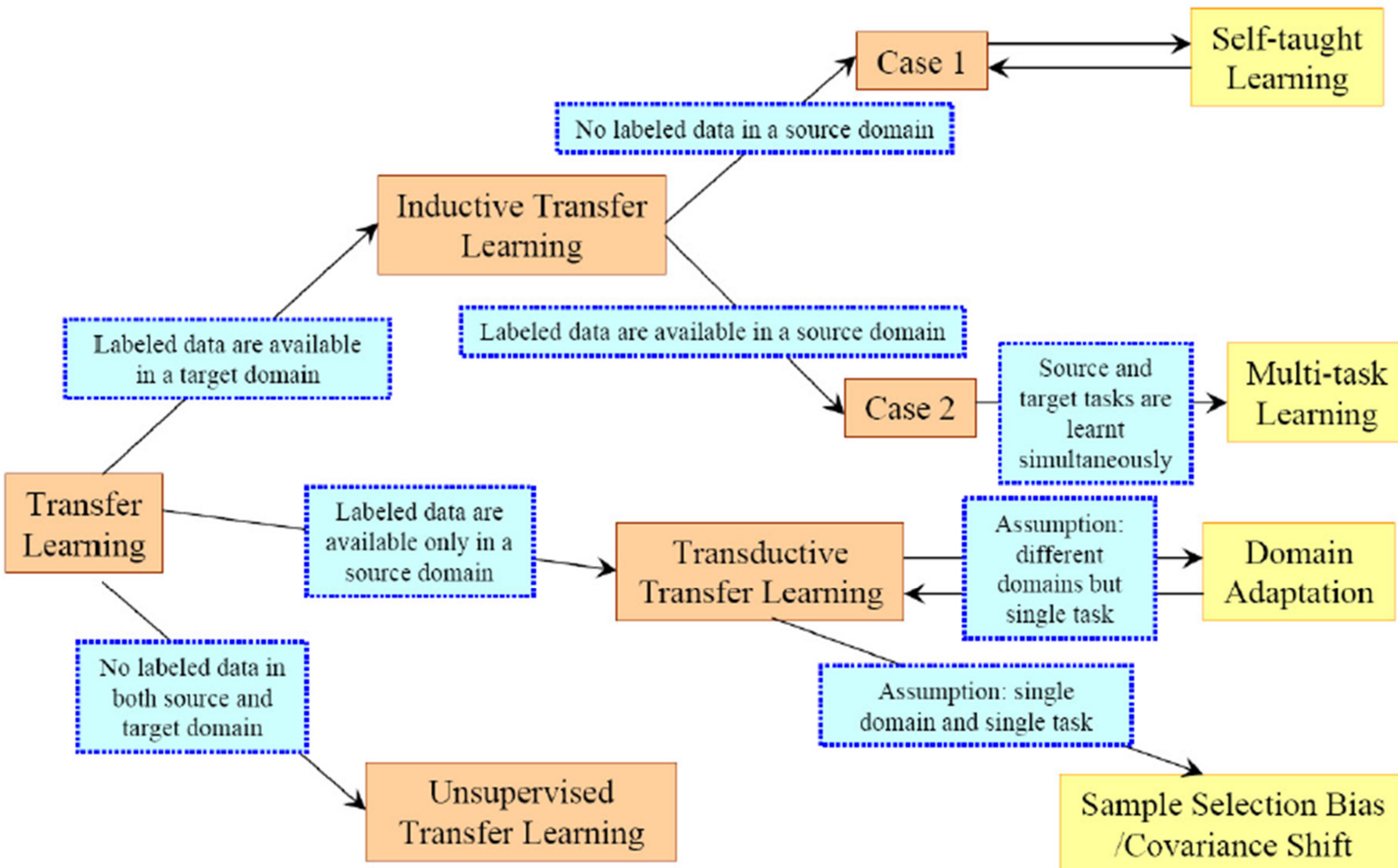
Notation and Definition of TL

- Notation
  - A **domain** $\mathcal{D} = \{\mathcal{X}, p(x)\}$
    - Feature space $\mathcal{X}$
    - Data distribution $p(x)$
  - A **task** $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$
    - Label space $\mathcal{Y}$
    - Objective predictive function $f(\cdot)$
- Definition
  - Given a **source domain** $\mathcal{D}_S$ with corresponding learning task $\mathcal{T}_S$ and a **target domain** $\mathcal{D}_T$ with corresponding learning task $\mathcal{T}_T$
  - **transfer learning** is the process of improving the target predictive function $f_T(\cdot)$ by using the related information from $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$ or $\mathcal{T}_S \neq \mathcal{T}_T$

Explanation

- $\mathcal{D}_S \neq \mathcal{D}_T$
  - $\mathcal{X}_S \neq \mathcal{X}_T$
    - Heterogeneous transfer learning
    - Two sets of documents are described in different languages
  - $P(X_S) \neq P(X_T)$
    - Domain adaptation
    - Two sets of documents focus on different topics
- $\mathcal{T}_S \neq \mathcal{T}_T$
  - $\mathcal{Y}_S \neq \mathcal{Y}_T$
    - Source has two classes: positive or negative; target adds one class: neutral
  - $P_S(y|x) \neq P_T(y|x)$
    - A word can have different meanings in two domains

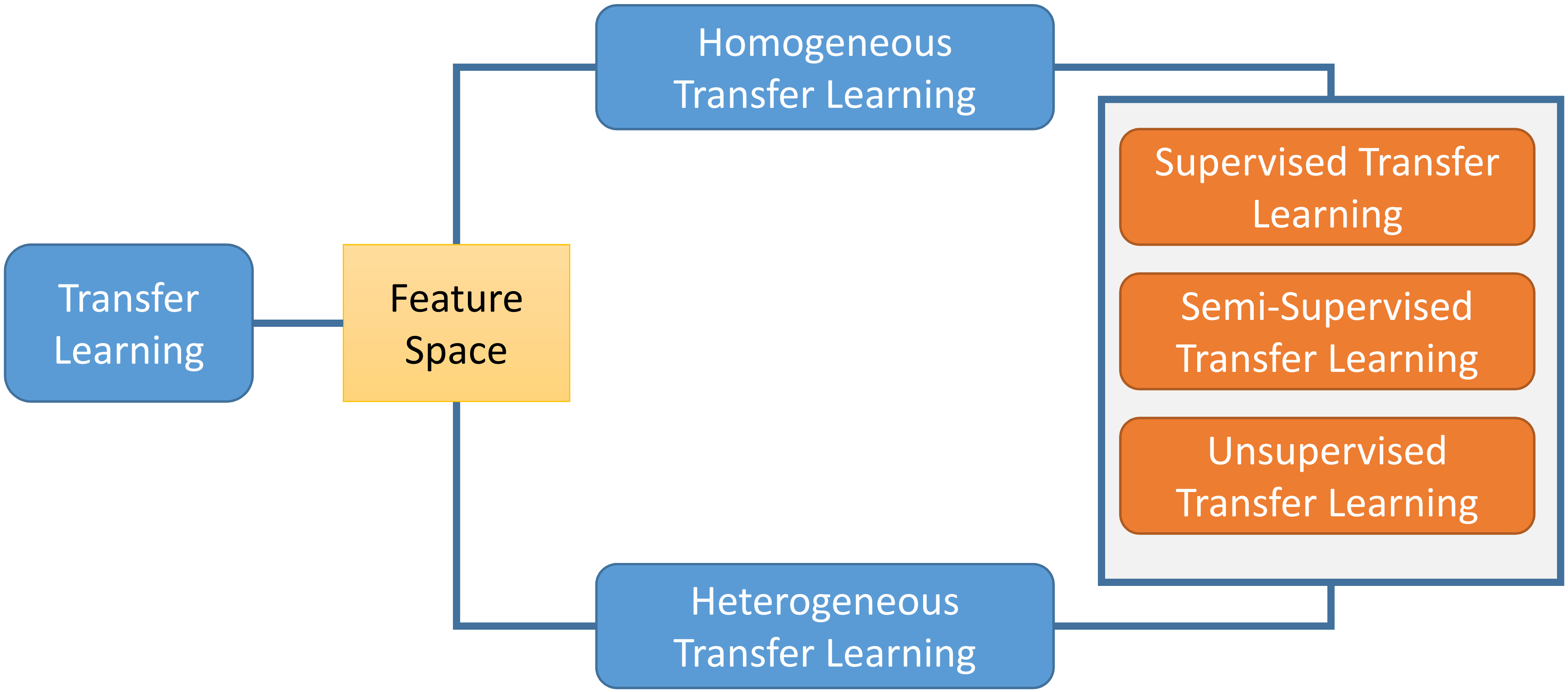
Categorization of Transfer Learning



Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *IEEE Transactions on knowledge and data engineering* 22.10 (2010): 1345-1359.

Transfer Learning Settings

- Homogeneous/heterogeneous transfer learning



Transfer Learning Methods

- Instance Transfer
  - Reweight instances of target data according to source
- Feature Transfer
  - Mapping features of source and target data in a common space
- Parameter Transfer
  - Learn target model parameters according to source model

Relational approaches are relatively unpopular, thus omitted in this talk

Transfer Learning Methods

- Instance Transfer
  - Reweight instances of target data according to source

- Feature Transfer
  - Mapping features of source and target data in a common space

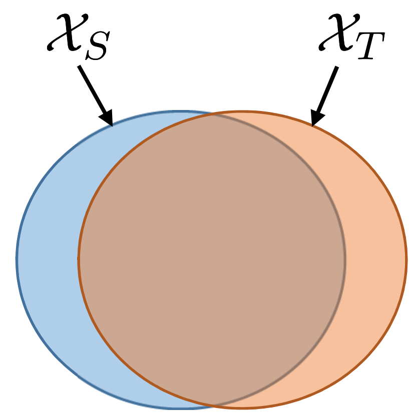
- Parameter Transfer
  - Learn target model parameters according to source model

Relational approaches are relatively unpopular, thus omitted in this talk

Instance-based Transfer Learning

- General assumption
  - Source and target domains have a lot of overlapping features or even share the same feature spaces

$$\mathcal{X}_S \simeq \mathcal{X}_T$$



  - Label space should be the same

$$\mathcal{Y}_S \simeq \mathcal{Y}_T$$

- Example applications
  - Electronic medical record across different departments
  - Sentiment analysis over different topics

Instance TL Case 1: Domain Adaption

- Problem setting
  - Given source domain labeled data $D_S = \{x_{S_i}, y_{S_i}\}_{i=1}^{n_S}$ and target domain data $D_T = \{x_{T_i}\}_{i=1}^{n_T}$
  - learn f_T such that the loss on target data is small

$$\sum_i \mathcal{L}(f_T(x_{T_i}), y_{T_i})$$

  - where y_{T_i} is unknown.
- Assumption
  - The same label space $\mathcal{Y}_S = \mathcal{Y}_T$
  - The same dependency $p(y_S|x_S) = p(y_T|x_T)$
  - (Almost) the same feature space $\mathcal{X}_S \simeq \mathcal{X}_T$
  - Different data distribution $p_S(x) \neq p_T(x)$

Importance Sampling for Domain Adaption

- Importance sampling

$$
\begin{aligned}
\theta^* &= \arg\min_\theta \mathbb{E}_{(x,y)\sim p_T}[\mathcal{L}(y, f_\theta(x))] \\
&= \arg\min_\theta \int_{(x,y)} p_T(x)\mathcal{L}(y, f_\theta(x))dx \\
&= \arg\min_\theta \int_{(x,y)} p_S(x)\frac{p_T(x)}{p_S(x)}\mathcal{L}(y, f_\theta(x))dx \\
&= \arg\min_\theta \mathbb{E}_{(x,y)\sim p_S}\Big[\frac{p_T(x)}{p_S(x)}\mathcal{L}(y, f_\theta(x))\Big]
\end{aligned}
$$

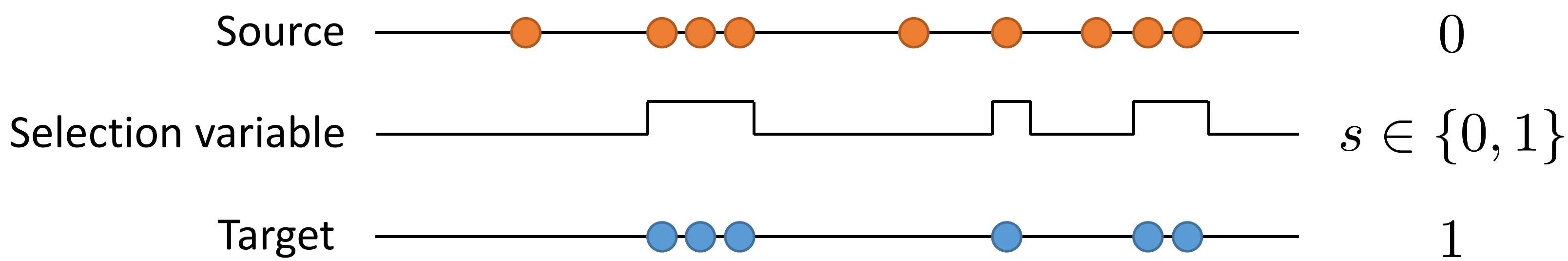
- Re-weight each instance by $\beta(x) = \dfrac{p_T(x)}{p_S(x)}$

Importance Sampling for Domain Adaption

- How to estimate $\beta(x) = \dfrac{p_T(x)}{p_S(x)}$
- A simple solution would be to first estimate $p_S(x)$ and $p_T(x)$ respectively, and then calculate $\beta(x)$
  - May suffer from huge variance problem
- A more practical solution is to estimate $\dfrac{p_T(x)}{p_S(x)}$ directly

Importance Sampling for Domain Adaption

- Imagine a rejection sampling process, and view the target domain as samples from the source domain



- Probabilistic density function (p.d.f.) relationship

$$p_T(x) \propto p_S(x)p(s=1|x)$$

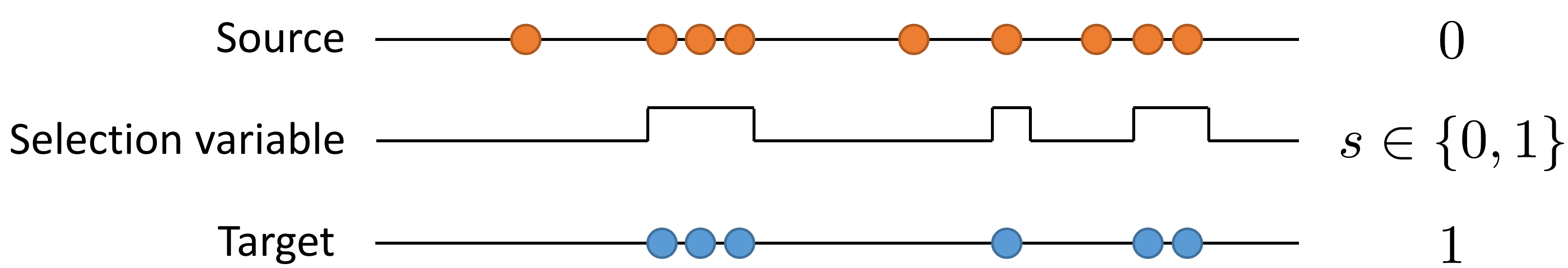
- And we estimate *p*(*s*=1|*x*) as a binary classification model

$$\beta(x) = \frac{p_T(x)}{p_S(x)} \propto p(s=1|x)$$

Zadrozny, Learning and Evaluating Classifiers under Sample Selection Bias, ICML 2004

Importance Sampling for Domain Adaption

- Imagine a rejection sampling process, and view the target domain as samples from the source domain



- Estimate $p(s=1|x)$ as a binary classification model
  - Label instance from the target domain as 1
  - Label instance from the source domain as 0

$$\beta(x) = \frac{p_T(x)}{p_S(x)} \propto p(s = 1|x)$$

Zadrozny, Learning and Evaluating Classifiers under Sample Selection Bias, ICML 2004

Importance Sampling for Domain Adaption

- How to estimate $\beta(x) = \dfrac{p_T(x)}{p_S(x)}$

- Build the estimator with a list of basis functions

$$\hat{\beta}(x) = \sum_{l=1}^{b} \alpha_l \psi_l(x)$$

- The estimated target p.d.f. $\hat{p}_T(x) = \hat{\beta}(x) p_S(x)$

- Minimize KL divergence

$$\min_{\{\alpha_l\}_{l=1}^{b}} \mathrm{KL}[p_T(x) \| \hat{p}_T(x)]$$

Sugiyama *et al*., Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation, NIPS 2007

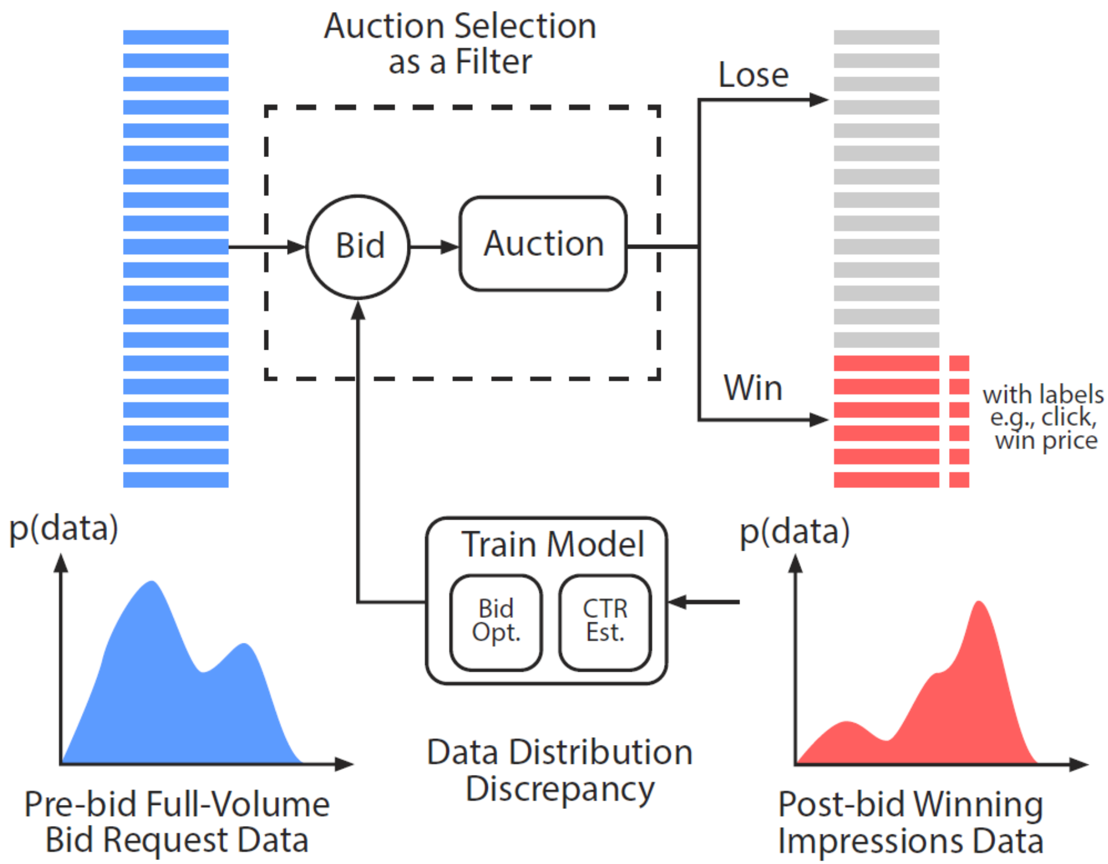
- Minimize squared error

$$\min_{\{\alpha_l\}_{l=1}^{b}} \int_x \left(\hat{\beta}(x) - \beta(x)\right)^2 p_S(x) dx$$

Kanamori et al., A Least-squares Approach to Direct Importance Estimation, JMLR 2009

Unbiased Training in Display Advertising

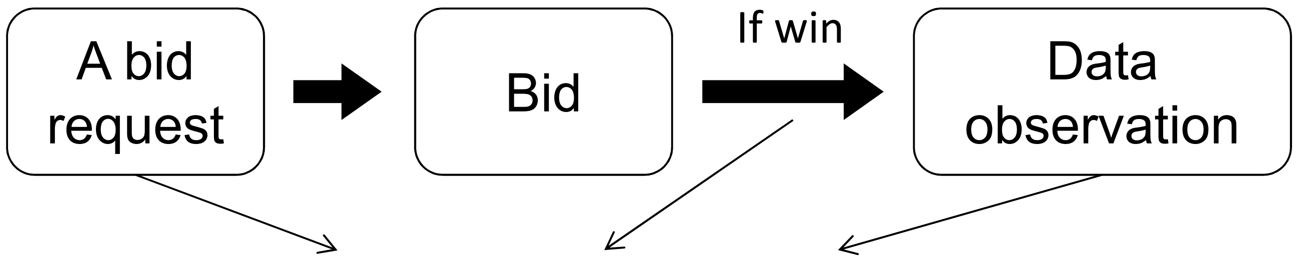
- In display advertising, the label data is observed by an advertiser only when she wins the auction, thus it is biased.



Weinan Zhang et al. Bid-aware Gradient Descent for Unbiased Learning with Censored Data in Display Advertising. KDD 16

Unbiased Learning Framework

- Data observation process



$$p(\boldsymbol{x})w(b_{\boldsymbol{x}}) \propto q(\boldsymbol{x})$$

- Importance sampling

$$\min_{\boldsymbol{\beta}} \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}[\mathcal{L}(y, f_{\boldsymbol{\beta}}(\boldsymbol{x}))] = \min_{\boldsymbol{\beta}} \mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}\Big[\frac{\mathcal{L}(y, f_{\boldsymbol{\beta}}(\boldsymbol{x}))}{w(b_{\boldsymbol{x}})}\Big]$$

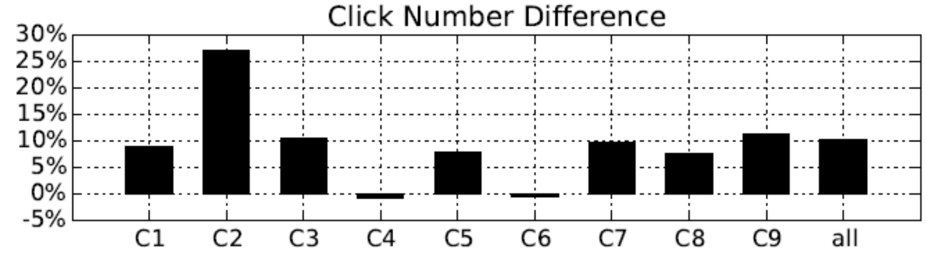
Weinan Zhang et al. Bid-aware Gradient Descent for Unbiased Learning with Censored Data in Display Advertising. KDD 16

Performance Comparison on Yahoo! DSP

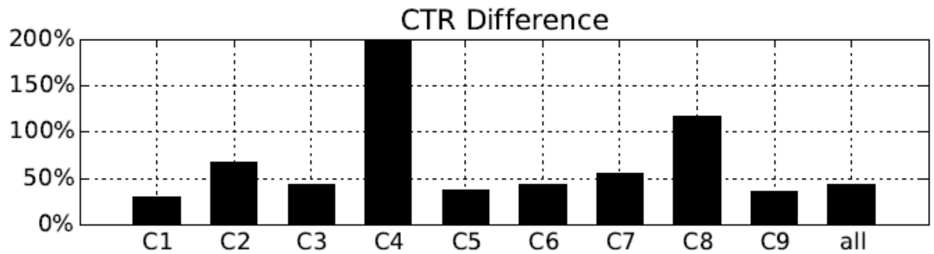
- A/B Testing on Yahoo! United States

2.97% AUC lift

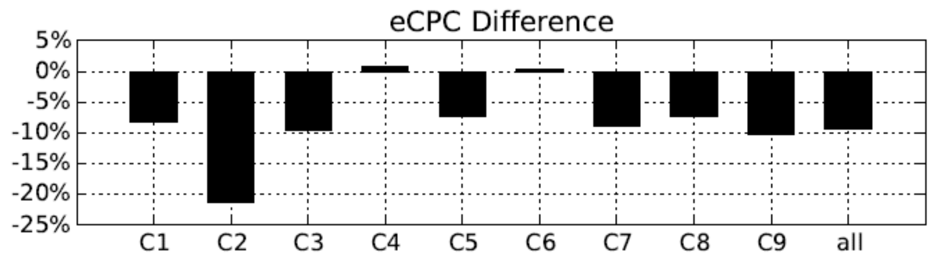
Camp.	BIAS AUC.	KMMP AUC	AUC Lift
C1	63.78%	64.12%	0.34%
C2	87.45%	88.58%	1.13%
C3	69.73%	75.52%	5.79%
C4	88.82%	89.55%	0.73%
C5	69.71%	72.29%	2.58%
C6	89.33%	90.70%	1.37%
C7	77.76%	78.92%	1.16%
C8	74.57%	76.98%	2.41%
C9	71.04%	73.12%	2.08%
all	73.48%	76.45%	2.97%



10.3% more clicks



42.8% higher CTR



9.3% lower eCPC

Weinan Zhang et al. Bid-aware Gradient Descent for Unbiased Learning with Censored Data in Display Advertising. KDD 16

Instance TL Case 2: Labels in 2 Domains

- Problem setting
  - Given source domain labeled data $D_S = \{x_{S_i}, y_{S_i}\}_{i=1}^{n_S}$
  - and very limited target domain data $D_T = \{x_{T_i}, y_{T_i}\}_{i=1}^{n_T}$
  - learn f_T such that the loss on target data is small

$$\sum_i \mathcal{L}(f_T(x_{T_i}), y_{T_i})$$

- Assumption
  - The same label space $\mathcal{Y}_S = \mathcal{Y}_T$
  - Different dependency $p(y_S|x_S) \neq p(y_T|x_T)$
  - (Almost) the same feature space $\mathcal{X}_S \simeq \mathcal{X}_T$
  - Different data distribution $p_S(x) \neq p_T(x)$

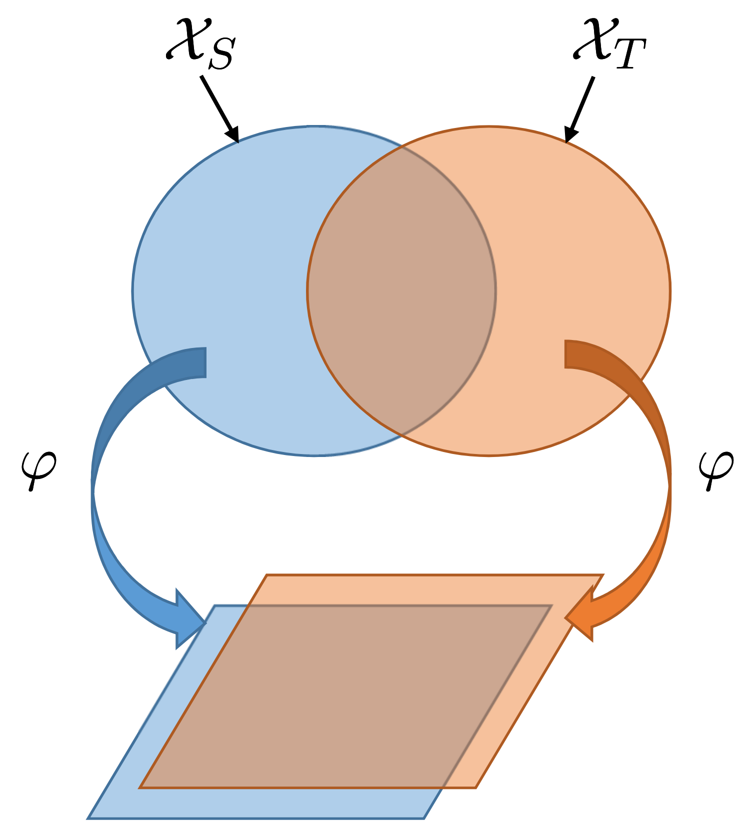
Transfer Learning Methods

- Instance Transfer
  - Reweight instances of target data according to source
- Feature Transfer
  - Mapping features of source and target data in a common space
- Parameter Transfer
  - Learn target model parameters according to source model

Relational approaches are relatively unpopular, thus omitted in this talk

Feature-based Transfer Learning

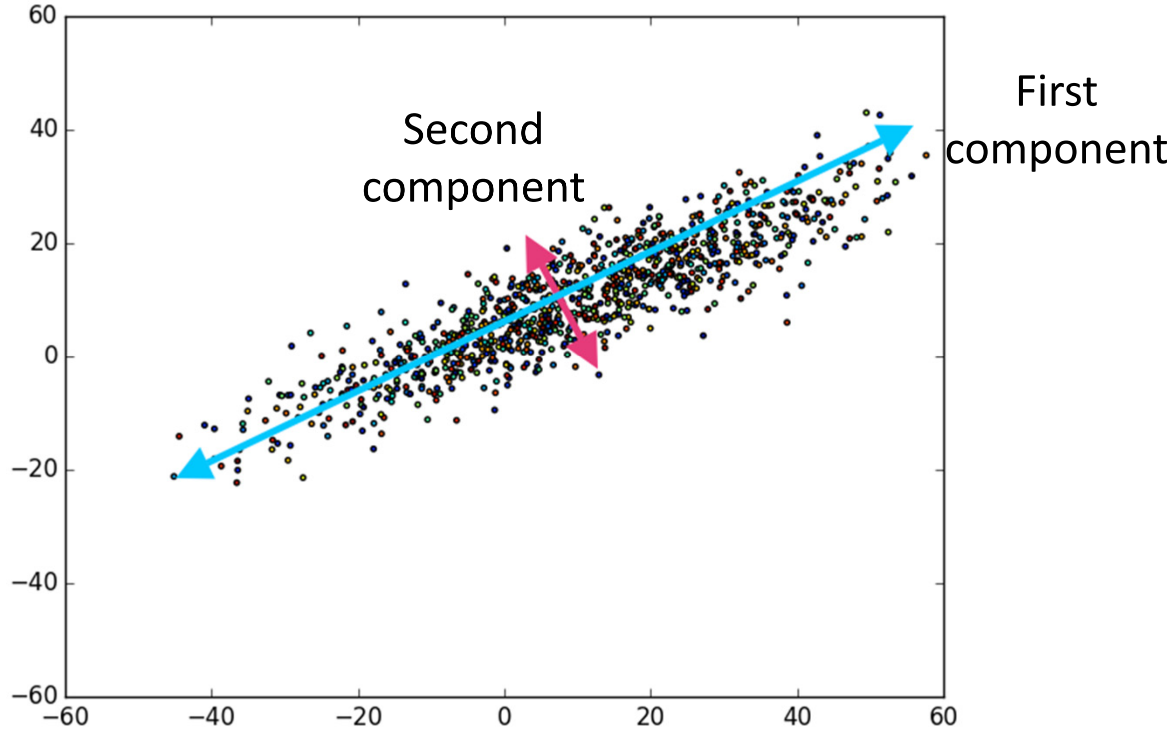
- When source and target domains only have some overlapping features
  - Lots of features only have support in either the source or the target domain
- Possible solutions
  - Encode application-specific knowledge
  - General approaches to learn the transformation φ



General Feature-Based TL Approach

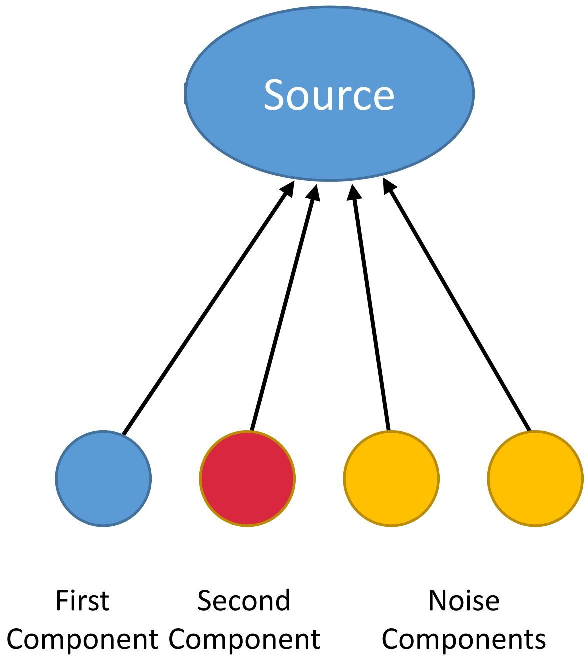
- Learning new data representations by minimizing the distance between two domain distributions
- Learning new data representations by multi-task learning
- Learning new data representations by self-taught learning

Principle Component Analysis (PCA)



- PCA uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components

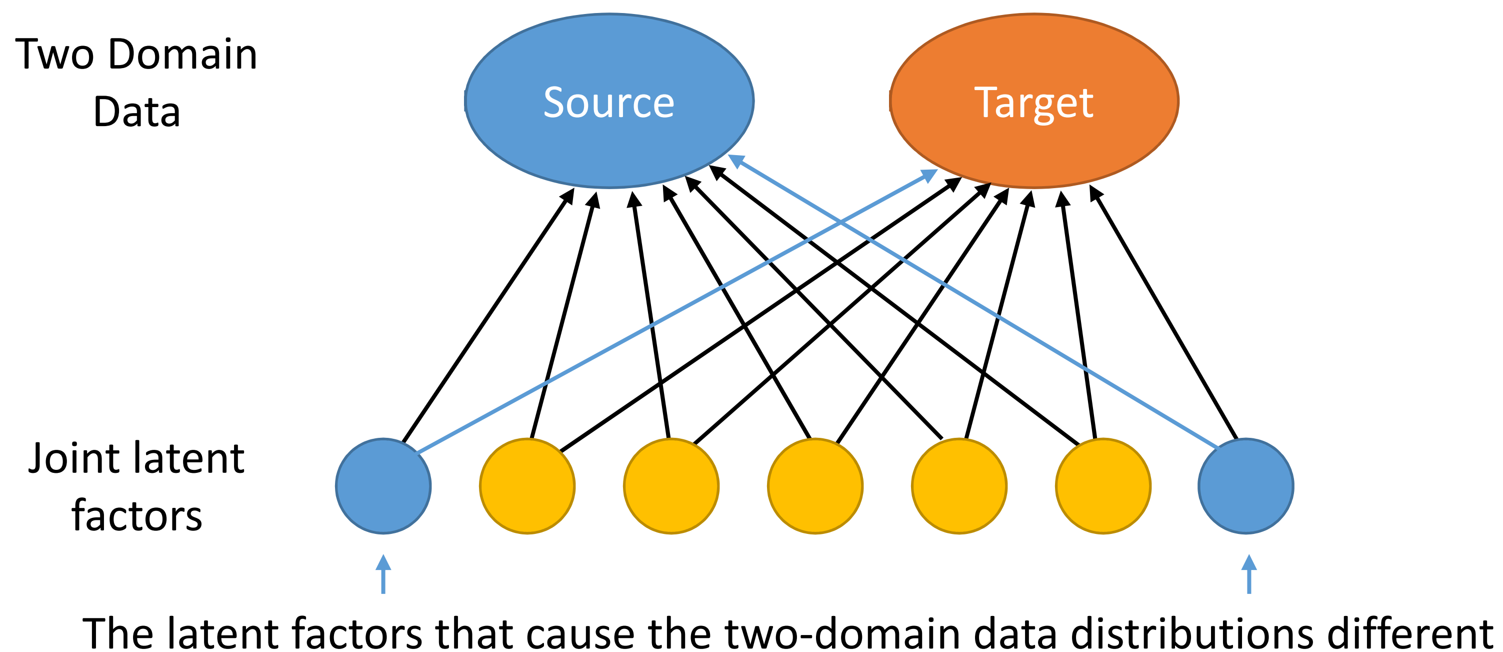
Principle Component Analysis (PCA)



- PCA uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components

Transfer Component Analysis

- Motivation
  - Minimize the distance between domain distributions by projecting data onto the learned transfer components



Pan, Sinno Jialin, et al. "Domain adaptation via transfer component analysis." *IEEE Transactions on Neural Networks* 22.2 (2011): 199-210.

Transfer Component Analysis

- Main idea
  - Learn φ to 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

$$
\begin{aligned}
\min_{\varphi} \quad & \mathrm{Dist}(\varphi(\mathbf{X}_S), \varphi(\mathbf{X}_T)) + \lambda\Omega(\varphi) \\
\text{s.t.} \quad & \text{constraints on } \varphi(\mathbf{X}_S) \text{ and } \varphi(\mathbf{X}_T)
\end{aligned}
$$

Transfer Component Analysis

- Maximum Mean Discrepancy (MMD)
  - Given the source and target domain data

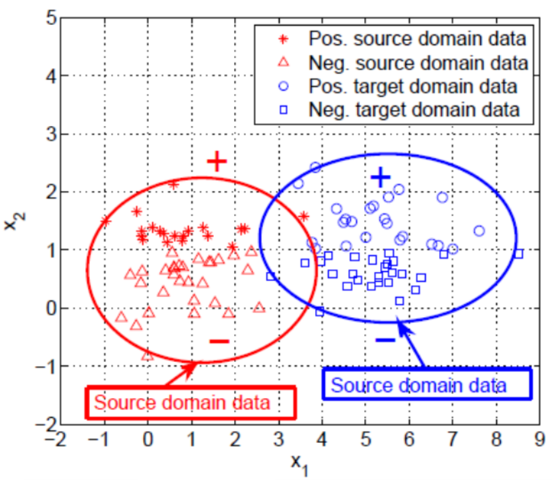
$$\mathbf{X}_S = \{x_{S_i}\}_{i=1}^{n_S} \qquad \mathbf{X}_T = \{x_{T_i}\}_{i=1}^{n_T}$$

drawn from $P_S(x)$ and $P_T(s)$ respectively

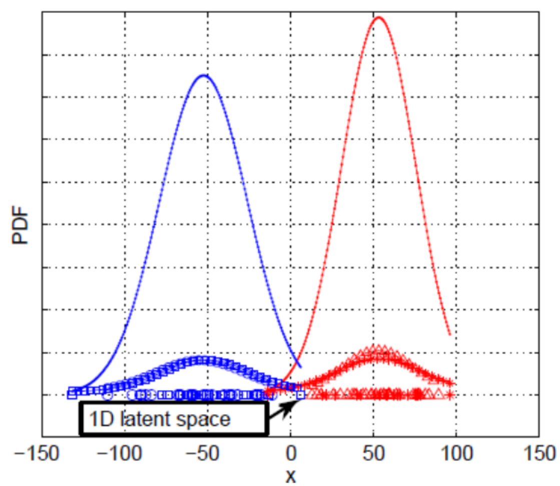
$$\mathrm{Dist}(\underbrace{\varphi}_{\text{Mapping}}(\mathbf{X}_S), \varphi(\mathbf{X}_T)) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \underbrace{\Phi}_{\text{Kernel function}}(\varphi(x_{S_i})) - \frac{1}{n_T} \sum_{i=1}^{n_T} \Phi(\varphi(x_{T_i})) \right\|_{\mathcal{H}}$$

Transfer Component Analysis

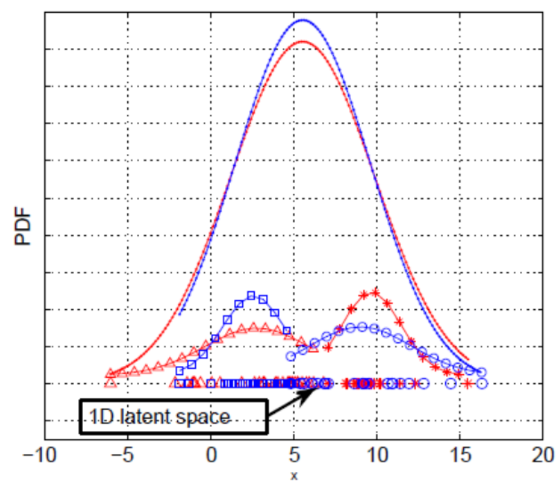
- An illustrative example Latent features learned by PCA and TCA



Original feature space



PCA



TCA

Transfer Learning Methods

- Instance Transfer
  - Reweight instances of target data according to source
- Feature Transfer
  - Mapping features of source and target data in a common space
- Parameter Transfer
  - Learn target model parameters according to source model

Relational approaches are relatively unpopular, thus omitted in this talk

Parameter based Transfer Learning

- The ϑ-parameterized function $f_\vartheta(x)$ learned on two domains

$$\theta_S^* = \arg\min_\theta \sum_{i=1}^{n_S} \mathcal{L}(y_{S_i}, f_\theta(x_{S_i})) + \lambda\Omega(\theta)$$

$$\theta_T^* = \arg\min_\theta \sum_{i=1}^{n_T} \mathcal{L}(y_{T_i}, f_\theta(x_{T_i})) + \lambda\Omega(\theta)$$

- Motivation
  - A well-trained model $f_{\theta_S^*}(x)$ has learned a lot of structure on the source domain.
  - If two tasks are related, this structure can be transferred to learn the model $f_{\theta_T^*}(x)$ on the target domain

Multi-Task or Collective Learning

- Minimize the joint loss on two tasks and the model parameters distance

$$\min_{\theta_S,\theta_T} \alpha \frac{1}{N_S}\sum_{i=1}^{N_S}\mathcal{L}(y_i, f_{\theta_S}(x_i)) + (1-\alpha)\frac{1}{N_T}\sum_{j=1}^{N_T}\mathcal{L}(y_j, f_{\theta_T}(x_j)) + \lambda\Omega(\theta_S,\theta_T)$$

Source task loss   Target task loss   Parameter distance

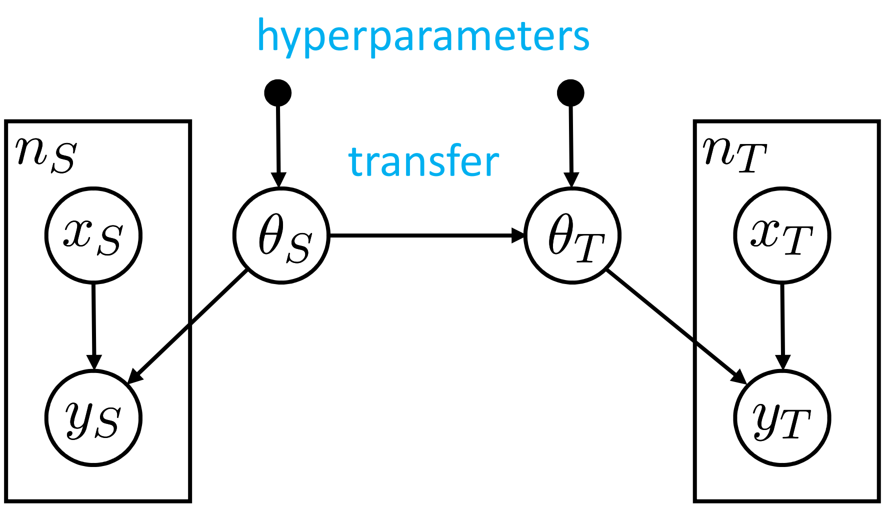
- Different parameter distance definitions

$$\Omega(\theta_S,\theta_T) = \|\theta_S - \theta_T\|^2$$

$$\Omega(\theta_S,\theta_T) = \sum_{t\in\{S,T\}} \|\theta_t - \frac{1}{2}\sum_{s\in\{S,T\}}\theta_s\|^2$$

Hierarchical Bayesian Network

- Idea: source domain parameters, regarded as random variables, act as the prior of the target domain parameters



Case Study: from web browsing to ad click

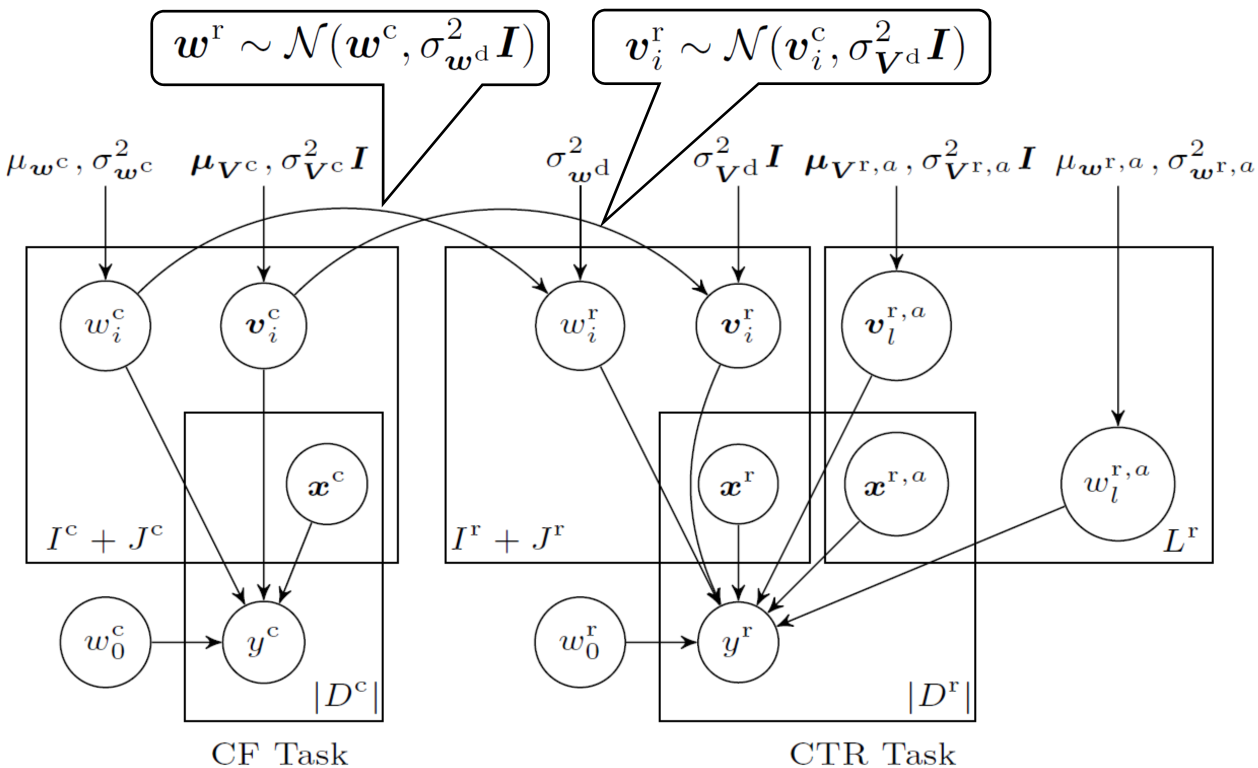
- Source task
  - Data: user browsed webpage ids
  - Task: predict whether a user likes a webpage
- Target task
  - Data: user browsed webpage ids
  - Task: predict whether a user likes to click an ad

$$\min_{\theta_S,\theta_T} \alpha \underbrace{\frac{1}{N_S}\sum_{i=1}^{N_S}\mathcal{L}(y_i, f_{\theta_S}(x_i))}_{\text{Logistic regression}} + (1-\alpha)\underbrace{\frac{1}{N_T}\sum_{j=1}^{N_T}\mathcal{L}(y_j, f_{\theta_T}(x_j))}_{\text{Logistic regression}} + \lambda\|\theta_S - \theta_T\|^2$$

[Perlich, Claudia, et al. "Machine learning for targeted display advertising: Transfer learning in action." *Machine learning* 95.1 (2014): 103-127.]

Case Study: from web browsing to ad click

- Illustrated in a hierarchical Bayesian graphical model



[Zhang, Weinan et al. Implicit Look-alike Modelling in Display Ads: Transfer Collaborative Filtering to CTR Estimation. ECIR 2016]

Heterogeneous TL

- Different feature space
- Examples
  - Cross-language document classification
  - Cross-system recommendation
- Approaches
  - Symmetric transformation mapping
  - Asymmetric transformation mapping

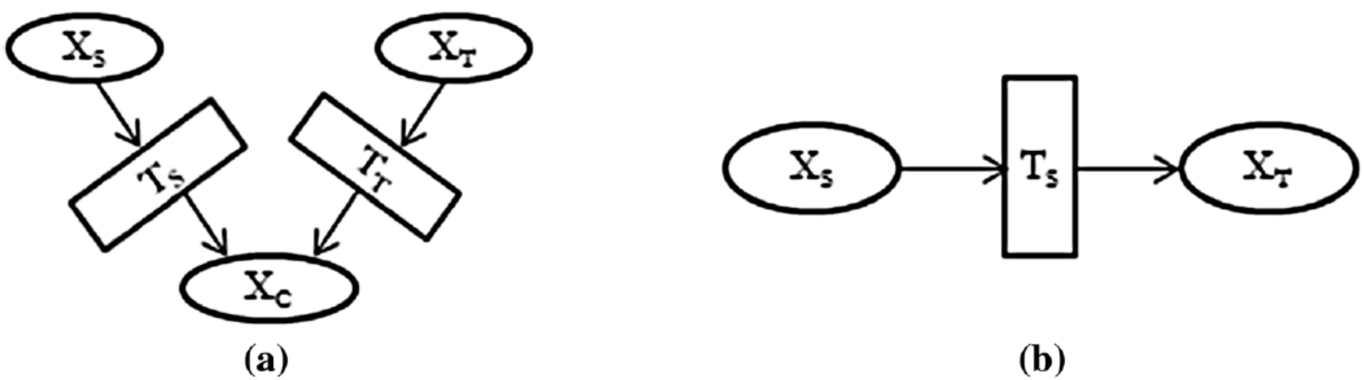
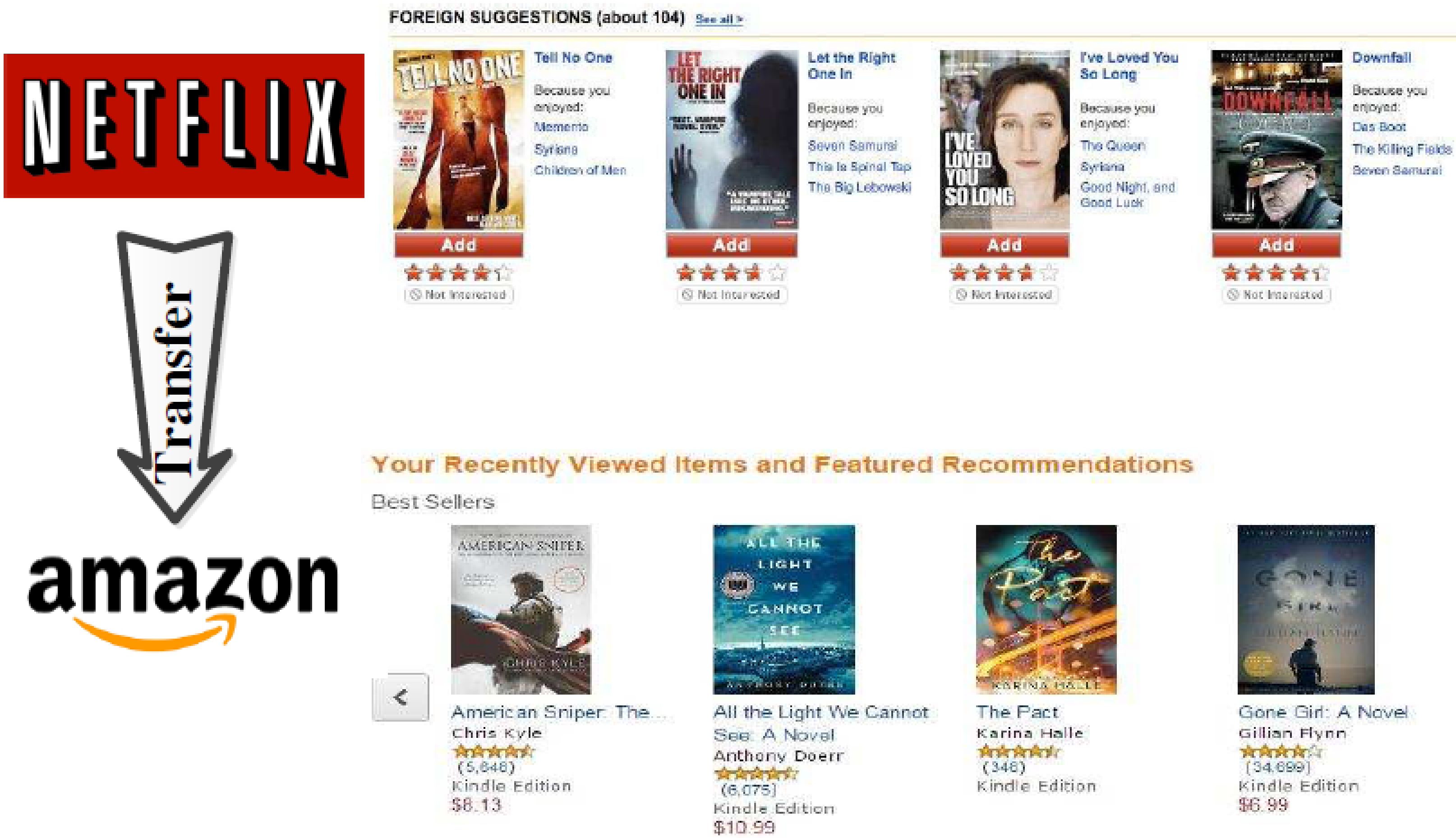
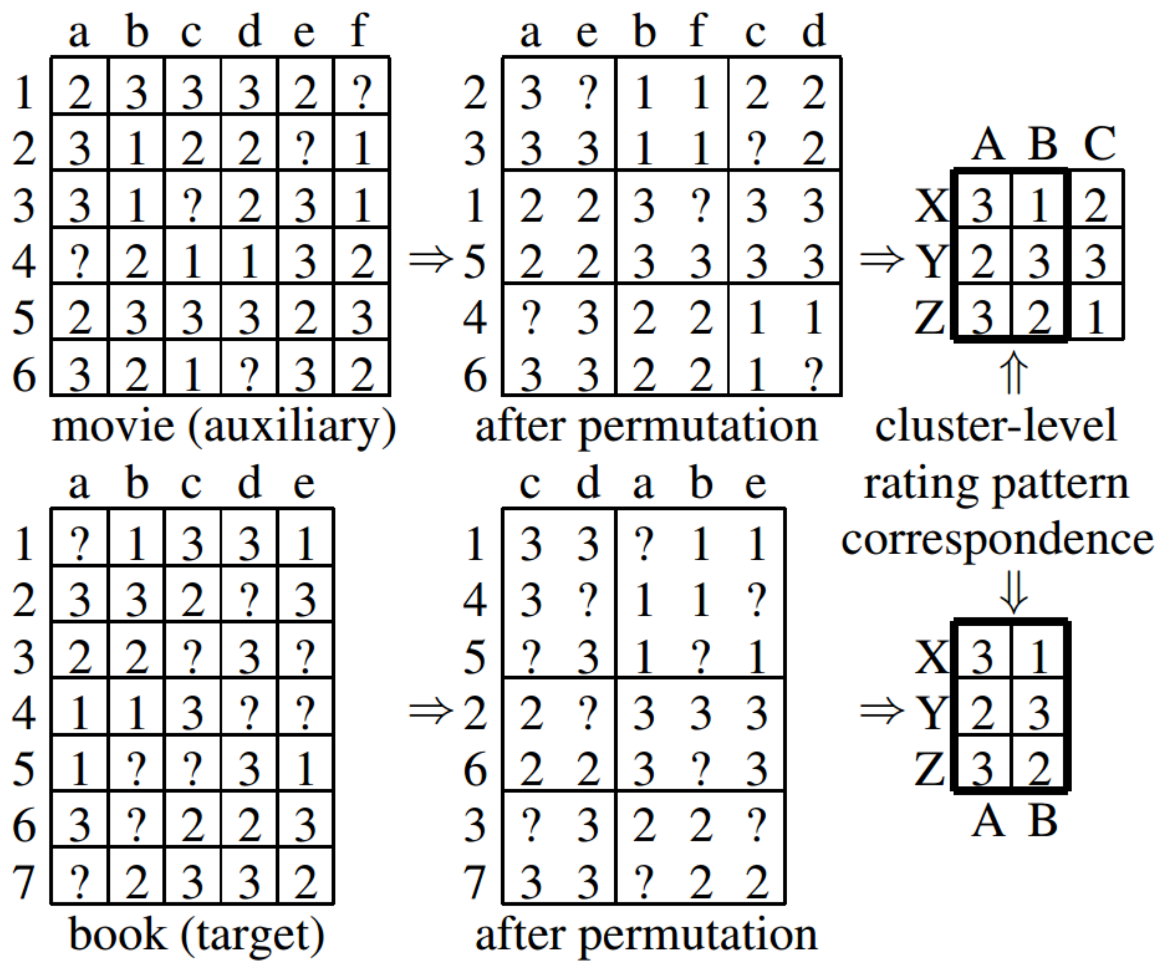


Fig. 1 **a** The symmetric transformation mapping (T_S and T_T) of the source (X_S) and target (X_T) domains into a common latent feature space. **b** The asymmetric transformation (T_T) of the source domain (X_S) to the target domain (X_T)

Cross-system Recommendation



Transfer Learning via CodeBook



Li, Bin, Qiang Yang, and Xiangyang Xue. "Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction." *IJCAI*. Vol. 9. 2009.

Transfer Learning via CodeBook

$$\underbrace{\begin{bmatrix} 1^* & 1 & 3 & 3 & 1 \\ 3 & 3 & 2 & 2^* & 3 \\ 2 & 2 & 3^* & 3 & 2^* \\ 1 & 1 & 3 & 3^* & 1^* \\ 1 & 1^* & 3^* & 3 & 1 \\ 3 & 3^* & 2 & 2 & 3 \\ 2^* & 2 & 3 & 3 & 2 \end{bmatrix}}_{\text{reconstructed}} \Leftarrow \overset{\mathbf{U}_{tgt}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} \times \underbrace{\begin{array}{c|ccc} & A & B & C \\ \hline X & 3 & 1 & 2 \\ Y & 2 & 3 & 3 \\ Z & 3 & 2 & 1 \end{array}}_{\text{codebook}} \times \overset{\mathbf{V}_{tgt}^{\top}}{\begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}$$

Table 1: MAE on MovieLens (average over 10 splits)

Training Set	Method	Given5	Given10	Given15
ML100	PCC	0.930	0.883	0.873
	CBS	0.874	0.845	0.839
	WLR	0.915	0.875	0.890
	CBT	**0.840**	**0.802**	**0.786**
ML200	PCC	0.905	0.878	0.878
	CBS	0.871	0.833	0.828
	WLR	0.941	0.903	0.883
	CBT	**0.839**	**0.800**	**0.784**
ML300	PCC	0.897	0.882	0.885
	CBS	0.870	0.834	0.819
	WLR	1.018	0.962	0.938
	CBT	**0.840**	**0.801**	**0.785**

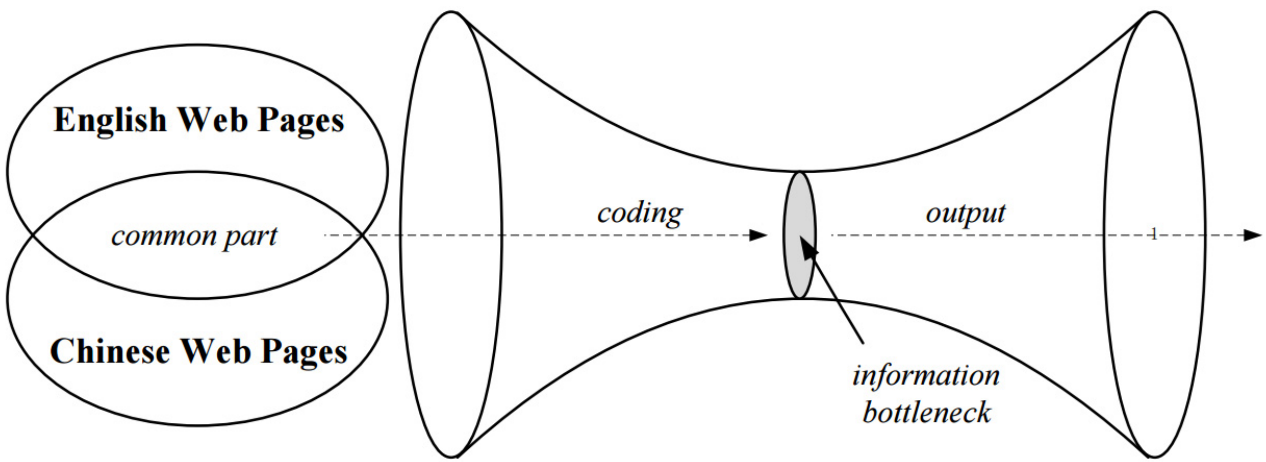
Table 2: MAE on Book-Crossing (average over 10 splits)

Training Set	Method	Given5	Given10	Given15
BX100	PCC	0.677	0.710	0.693
	CBS	0.664	0.655	0.641
	WLR	1.170	1.182	1.174
	CBT	**0.614**	**0.611**	**0.593**
BX200	PCC	0.687	0.719	0.695
	CBS	0.661	0.644	0.630
	WLR	0.965	1.024	0.991
	CBT	**0.614**	**0.600**	**0.581**
BX300	PCC	0.688	0.712	0.682
	CBS	0.659	0.655	0.633
	WLR	0.842	0.837	0.829
	CBT	**0.605**	**0.592**	**0.574**

Li, Bin, Qiang Yang, and Xiangyang Xue. "Can Movies and Books Collaborate? Cross-Domain Collaborative Filtering for Sparsity Reduction." *IJCAI*. Vol. 9. 2009.

Cross-Language Text Classification

- A large number of labeled English webpages
- A small number of labeled Chinese webpages
- Solution: information bottleneck



Ling, Xiao, et al. "Can chinese web pages be classified with english data source?." *WWW* 2008.