An Overview of Transfer Learning

Qiang Yang,
Huawei Noah's Ark Lab and HKUST

Thanks: Sinno Pan
NTU and I2R Singapore

http://www.cse.ust.hk/TL

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
Different fields

- Transfer learning for reinforcement learning.

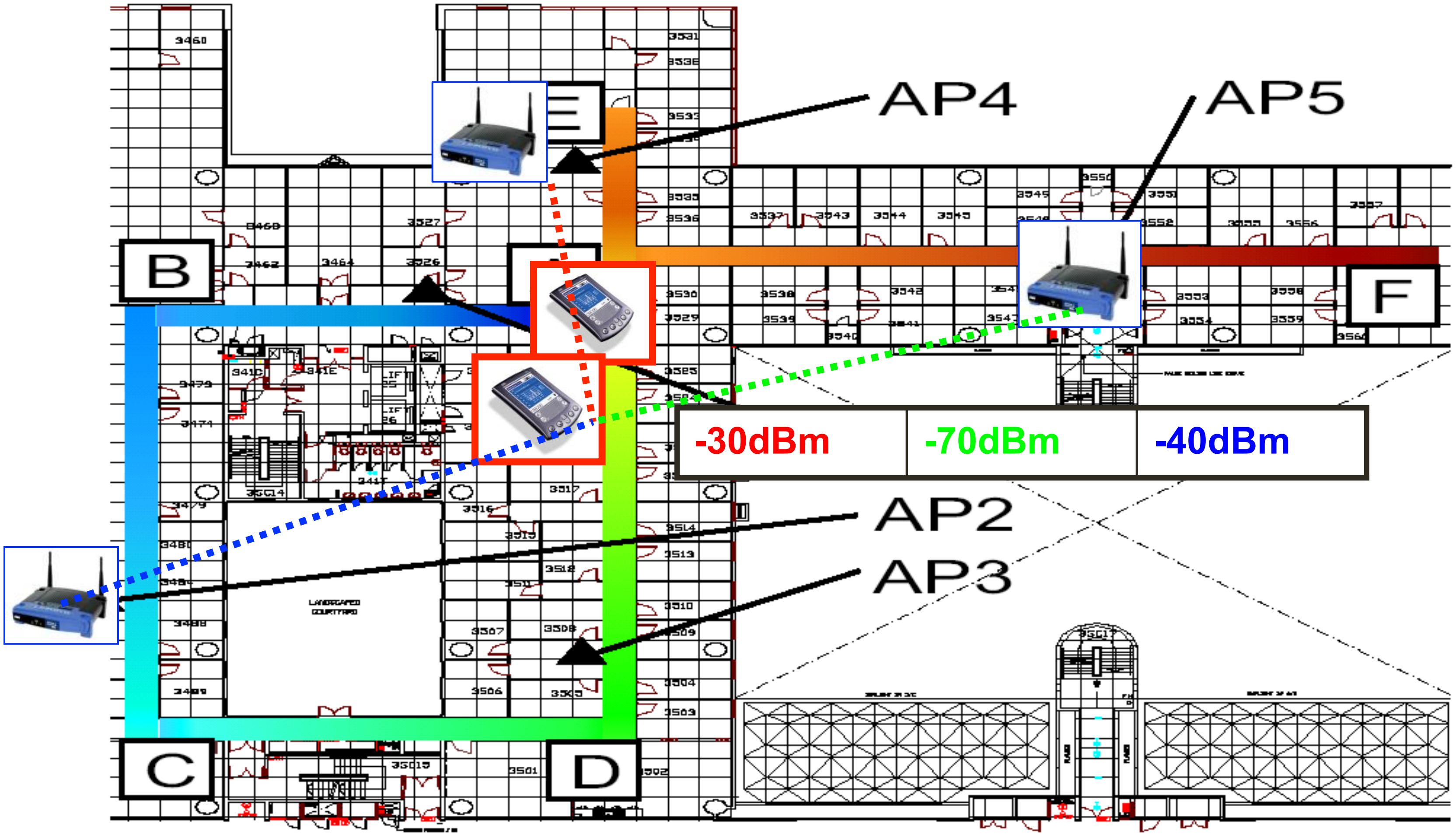
[Taylor and Stone, Transfer Learning for Reinforcement Learning Domains: A Survey, JMLR 2009]

- Transfer learning for classification, and regression problems.

Focus!

[Pan and Yang, A Survey on Transfer Learning, IEEE TKDE 2010]

Motivating Example I:
Indoor WiFi localization

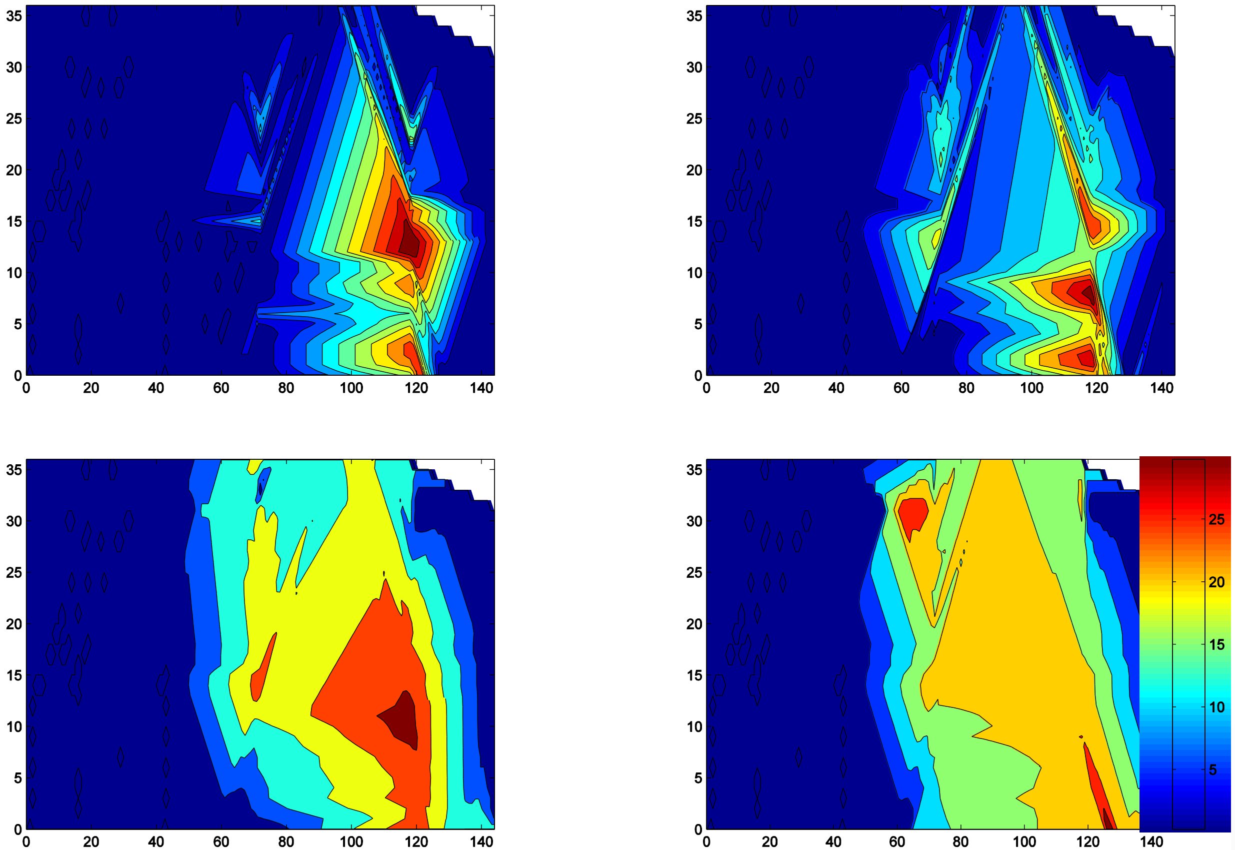


Difference between Domains

Time Period A | **Time Period B**

Device A

Device B



Indoor WiFi Localization (cont.)

Average Error Distance

Training

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

Localization model

Test

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

~ 1.5 meters

Device A

Device A

Drop!

Training

S=(-33dbm, .., -82dbm), L=(1, 3)
…
S=(-57dbm, .., -63dbm), L=(10, 23)

Localization model

Test

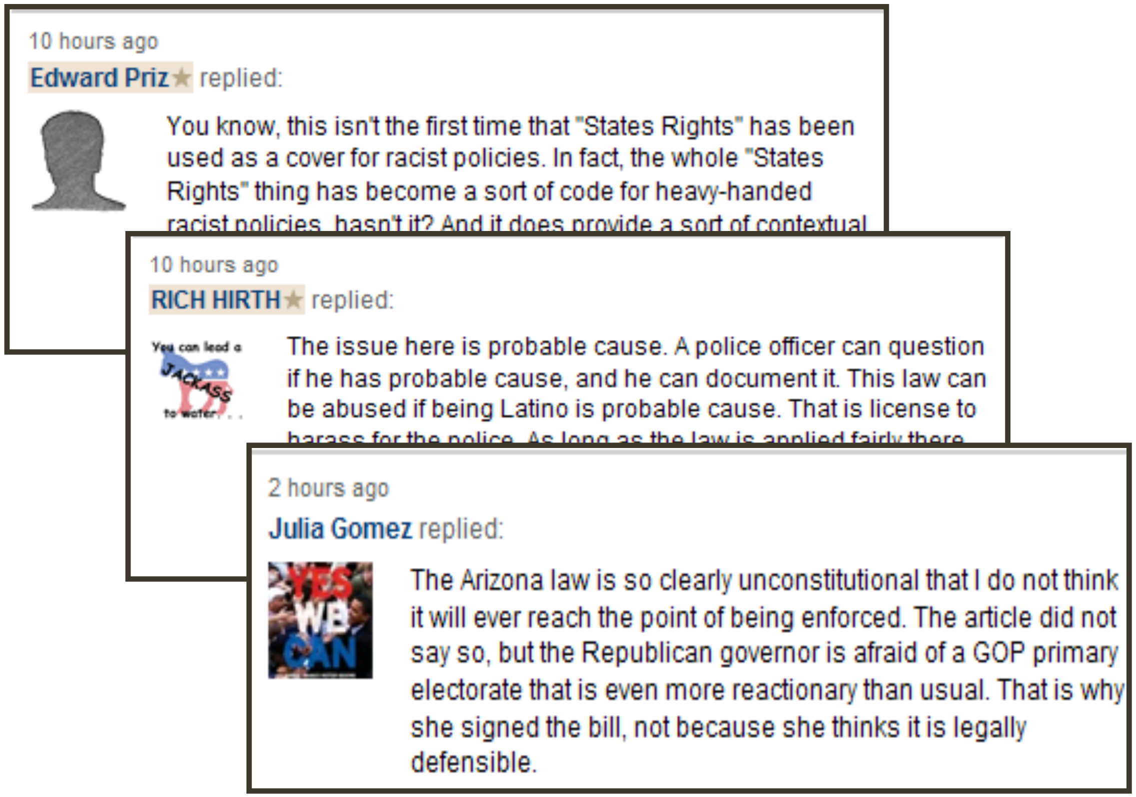
S=(-37dbm, .., -77dbm)
S=(-41dbm, .., -83dbm)
…
S=(-49dbm, .., -34dbm)
S=(-61dbm, .., -28dbm)

~10 meters

Device B

Device A

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

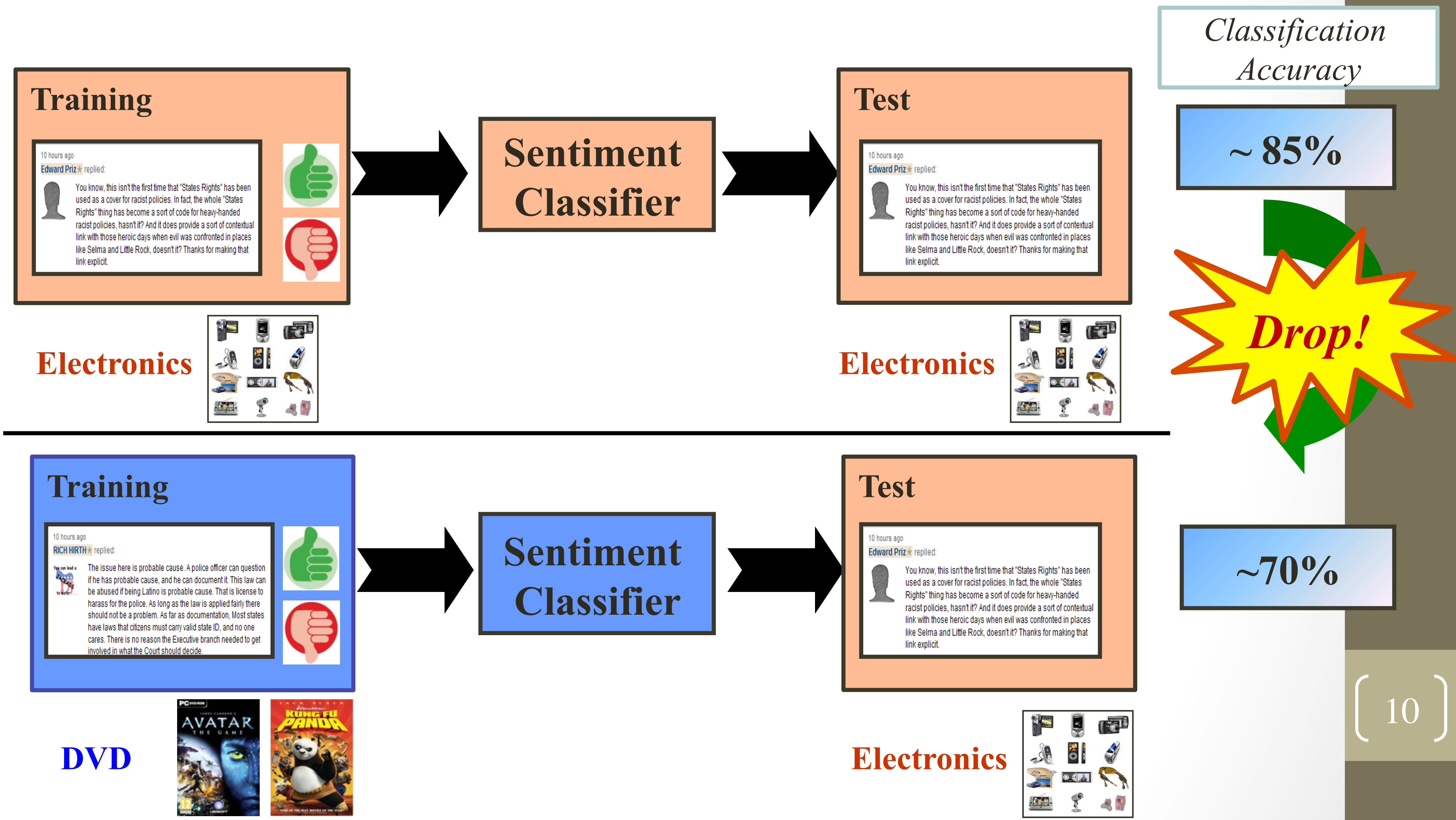
Julia Gomez replied:

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.

Difference between Domains

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

Sentiment Classification



A Major Assumption in Traditional Machine Learning

- Training and future (test) data come from the same domain, which implies
  - Represented in the same feature space.
  - 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:
  - Different predictive distributions, or different label spaces:

$$\mathcal{X}_S \neq \mathcal{X}_T, \text{ or } P_S(x) \neq P_T(x)$$

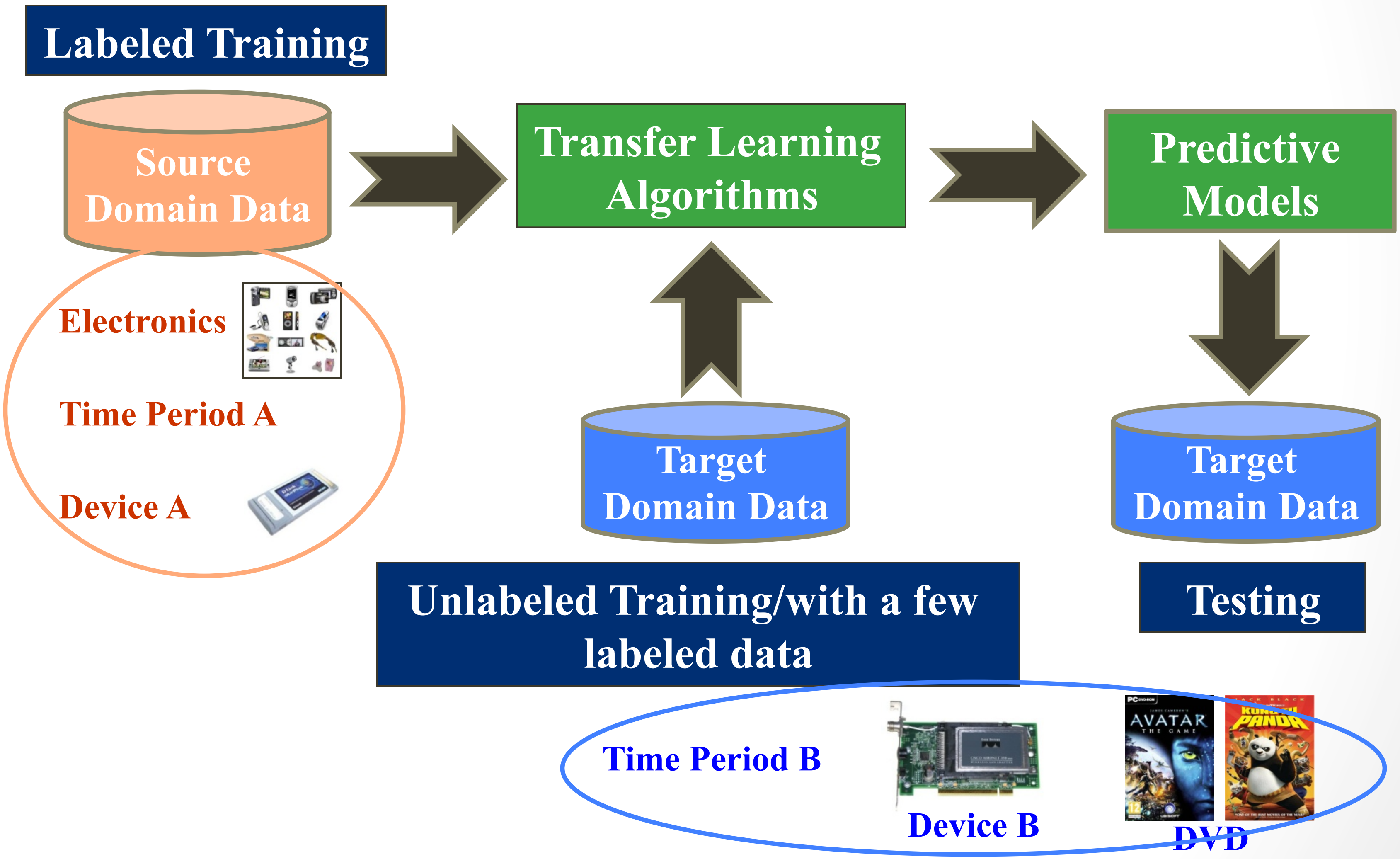
$$\mathcal{Y}_S \neq \mathcal{Y}_T, \text{ or } f_S \neq f_T \ (P_S(y|x) \neq P_T(y|x))$$

How to Build Systems on Each Domain of Interest

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

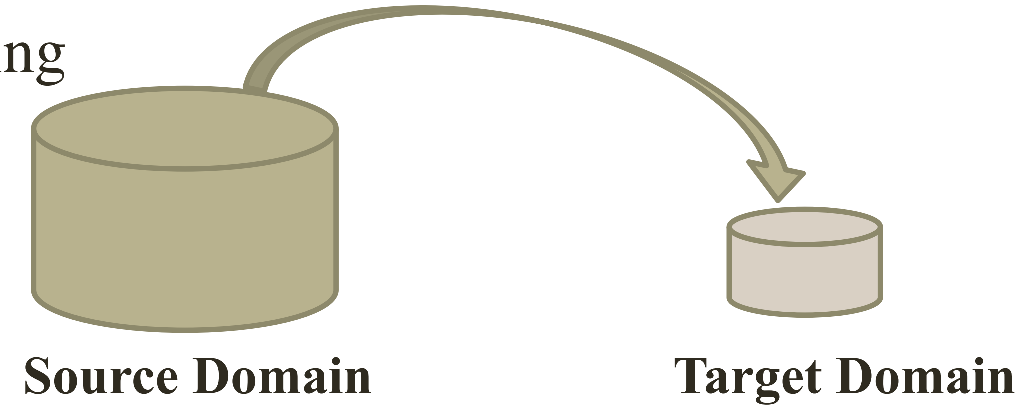
- Reuse common knowledge extracted from existing systems?
  - More practical!

The Goal of Transfer Learning

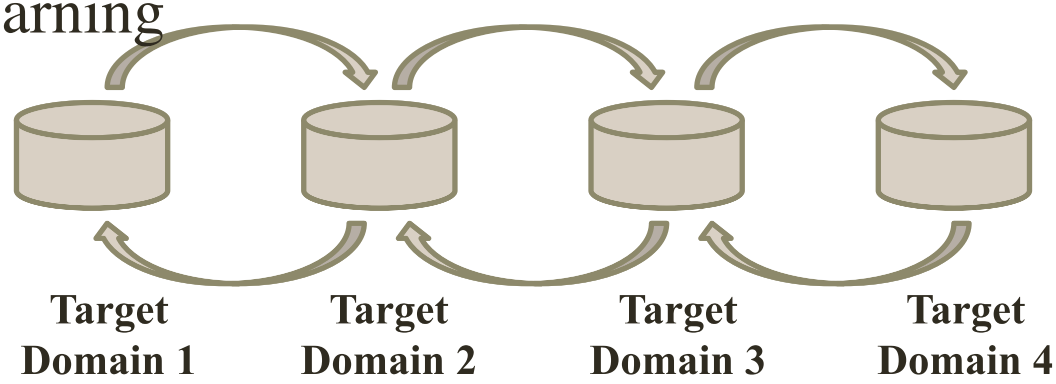


Transfer Learning v.s. Multi-task Learning

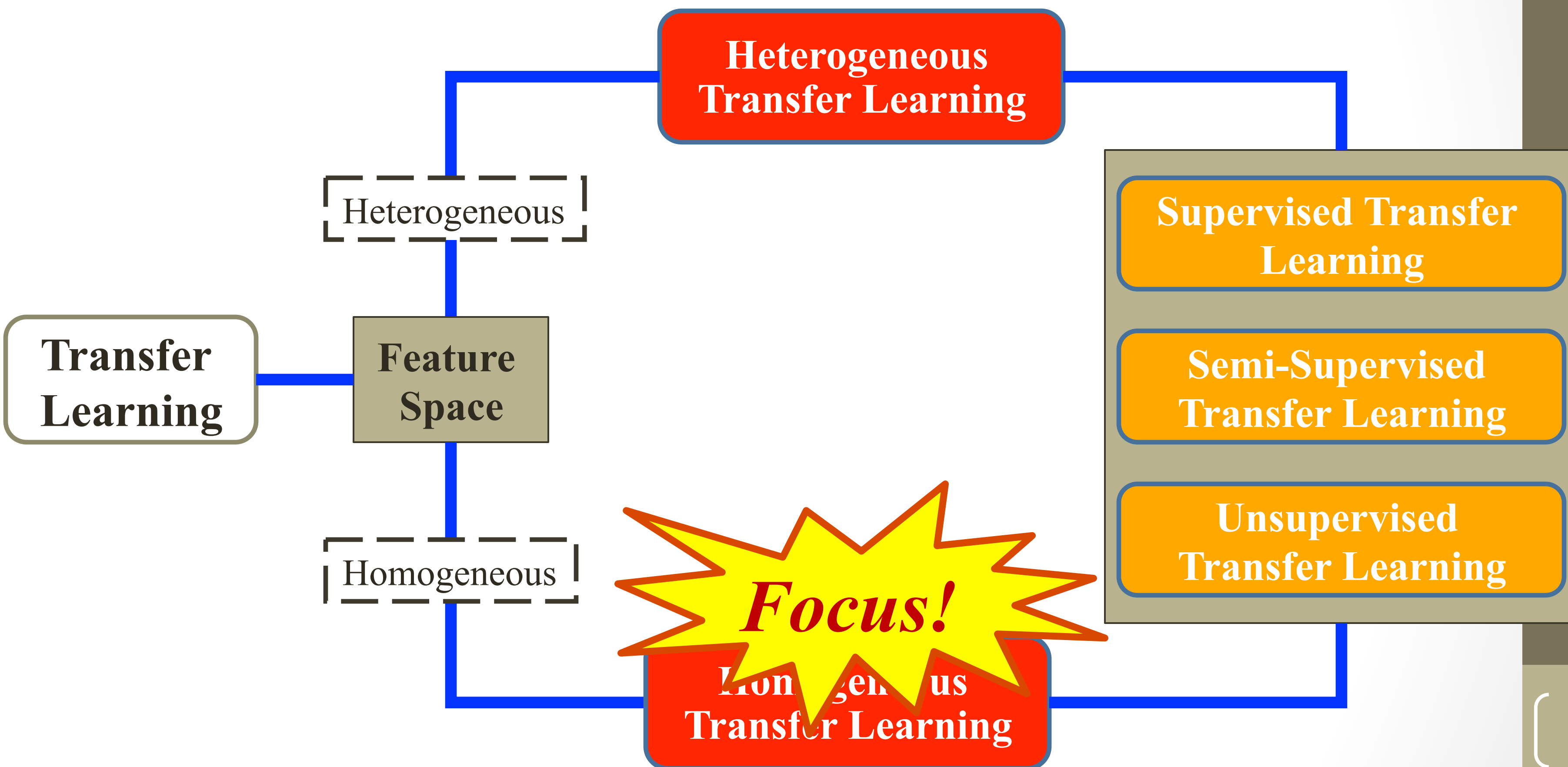
Transfer learning



Multi-task learning



Transfer Learning Settings



Transfer Learning Approaches

Instance-based Approaches

Feature-based Approaches

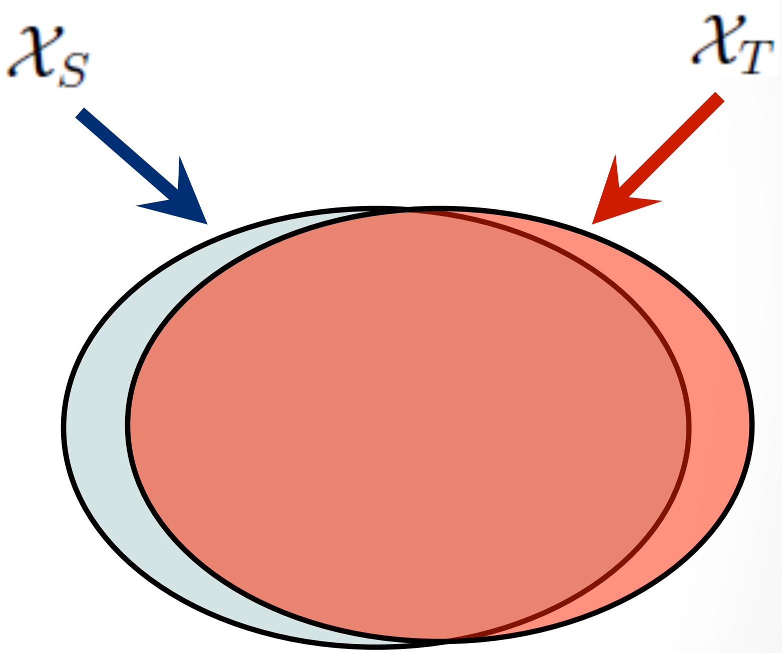
Parameter-based Approaches

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 $\mathbf{D}_S = \{x_{S_i}, y_{S_i}\}_{i=1}^{n_S}$, $\mathbf{D}_T = \{x_{T_i}\}_{i=1}^{n_T}$,

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 $\mathbf{D}_S = \{x_{S_i}, y_{S_i}\}_{i=1}^{n_S}$,

$\mathbf{D}_T = \{x_{T_i}, y_{T_i}\}_{i=1}^{n_T}$, $n_T \ll n_S$,

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,

$$
\begin{aligned}
\boxed{\theta^* \;=\; \arg\min \mathbb{E}_{(x,y)\sim P_T}\left[l(x,y,\theta)\right]} & \\
&= \arg\min \mathbb{E}_{(x,y)\sim P_T}\left[\frac{P_S(x,y)}{P_S(x,y)} l(x,y,\theta)\right] \\
&= \arg\min \int_y \int_x P_T(x,y)\left(\frac{P_S(x,y)}{P_S(x,y)} l(x,y,\theta)\right) dxdy \\
&= \arg\min \int_y \int_x P_S(x,y)\left(\frac{P_T(x,y)}{P_S(x,y)} l(x,y,\theta)\right) dxdy \\
&= \boxed{\arg\min \mathbb{E}_{(x,y)\sim P_S}\left[\frac{P_T(x,y)}{P_S(x,y)} l(x,y,\theta)\right]}
\end{aligned}
$$

Instance-based Approaches

Case I (cont.)

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

$$\begin{aligned}
\theta^* &= \arg\min \mathbb{E}_{(x,y)\sim P_S}\left[\frac{P_T(x,y)}{P_S(x,y)} l(x,y,\theta)\right] \\
&= \arg\min \mathbb{E}_{(x,y)\sim P_S}\left[\frac{P_T(x)P_T(y|x)}{P_S(x)P_S(y|x)} l(x,y,\theta)\right] \\
&= \boxed{\arg\min \mathbb{E}_{(x,y)\sim P_S}\left[\frac{P_T(x)}{P_S(x)} l(x,y,\theta)\right]}
\end{aligned}$$

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

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

Instance-based Approaches

Case I (cont.)

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

A simple solution is to first estimate $P_T(x),\ P_S(x)$, respectively,

and calculate $\dfrac{P_T(x)}{P_S(x)}$. 

An alterative solution is to estimate $\dfrac{P_T(x)}{P_S(x)}$ directly. 

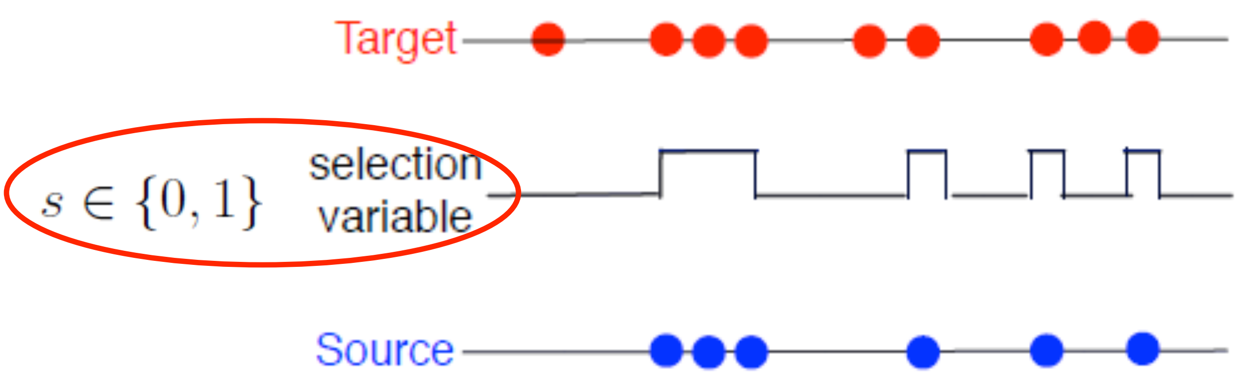
Correcting Sample Selection Bias / Covariate Shift

[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

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

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

[Zadrozny, ICML-04]

➢ Label instances from the source domain with label 1
➢ Label instances from the target domain with label 0
➢ Train a binary classifier

Instance-based Approaches

Kernel mean matching (KMM)

Maximum Mean Discrepancy (MMD)

Given $\mathbf{X}_S = \{x_{S_i}\}_{i=1}^{n_S}$, $\mathbf{X}_T = \{x_{T_i}\}_{i=1}^{n_T}$, drown from $P_S(x)$ and $P_T(x)$, respectively,

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

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

Instance-based Approaches

Kernel mean matching (KMM) (cont.)

[Huang *etal*., NIPS-06]

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

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

The required optimization is a simple QP problem

Instance-based Approaches

Direct density ratio estimation

Consider $\beta(x) = \frac{P_T(x)}{P_S(x)}$ to be a function, which can be approximated by

$$\widetilde{\beta}(x) = \sum_{\ell=1}^{b} \alpha_\ell \psi_\ell(x),$$

then $P_T(x)$ can be approximated by $\widetilde{P}_T(x) = \widetilde{\beta}(x) P_S(x)$

KL divergence loss

$$\underset{\{\alpha_\ell\}_{\ell=1}^{b}}{\arg\min} \, \mathrm{KL}[P_T(x) || \widetilde{P}_T(x)]$$

[Sugiyama *etal*., NIPS-07]

Least squared loss

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

[Kanamori *etal*., 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 based on their contributions to the classification accuracy of the learning problem in the target domain

Instance-based Approaches

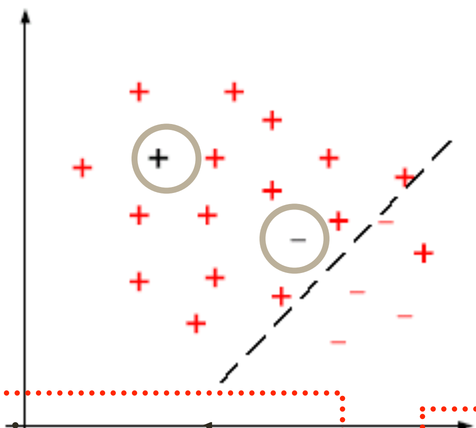
Case II (cont.)

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

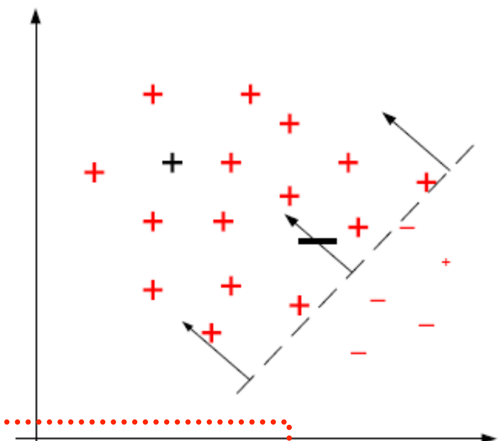
Instance-transfer Approaches

[Wu and Dietterich ICML-04]
[J.Jiang and C. Zhai, ACL 2007]
[Dai, Yang et al. ICML-07]

Uniform weights



Correct the decision boundary by re-weighting



Loss function on the target domain data

Loss function of source domain data

Regularization term

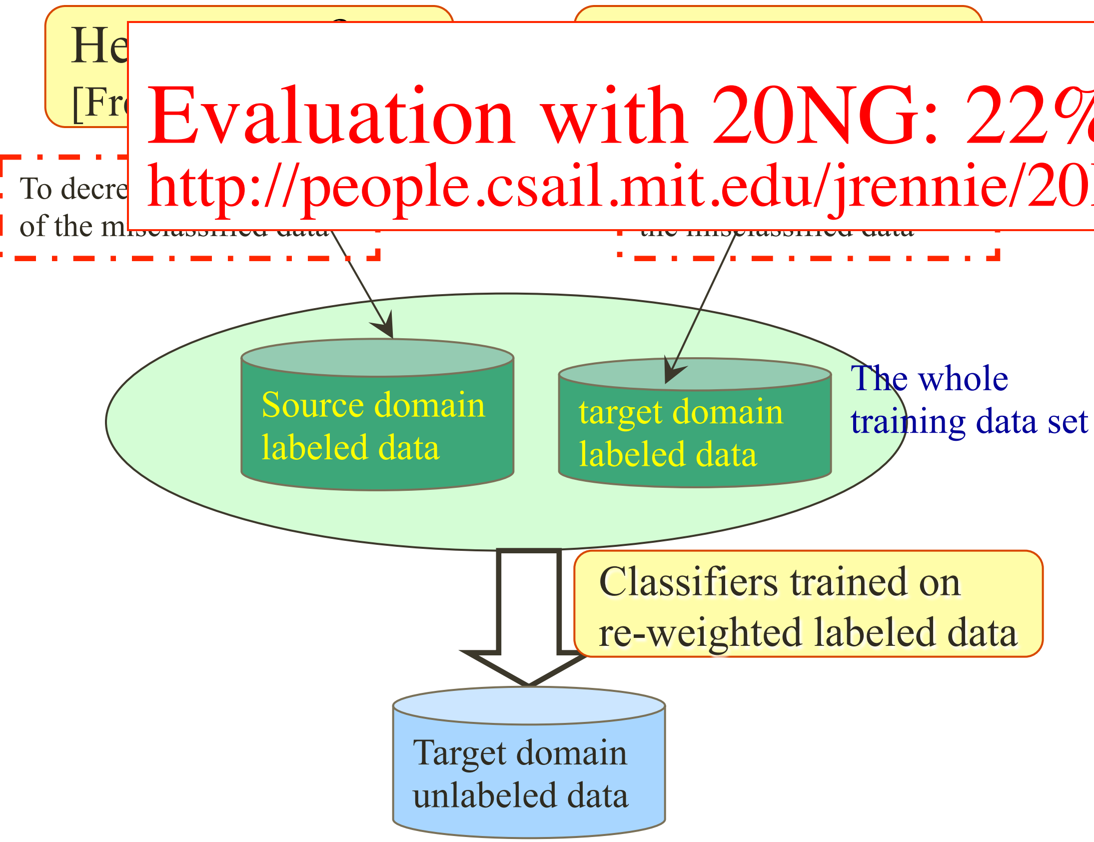
$$J(h) = \sum_{i}^{n_T} L(h(x_{T_i}), y_{T_i}) + \lambda \sum_{j}^{n_S} L(h(x_{S_j}), y_{T_j}) + R(h)$$

TrAdaBoost [Dai, Yang et al. ICML-07]

- Misclassified examples:

  data

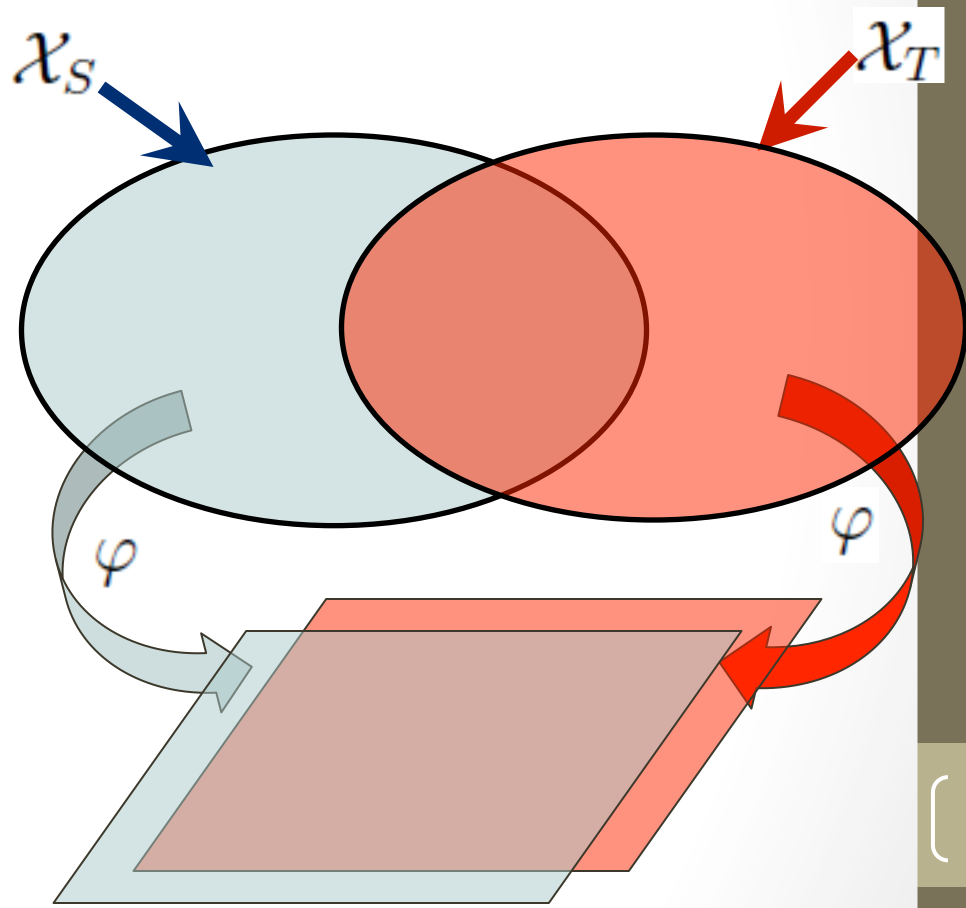
  – decrease the weights of the misclassified source data



Evaluation with 20NG: 22%→8%
http://people.csail.mit.edu/jrennie/20Newsgroups/

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 φ ?

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

- Solution 2: General approaches to learning the transformation.

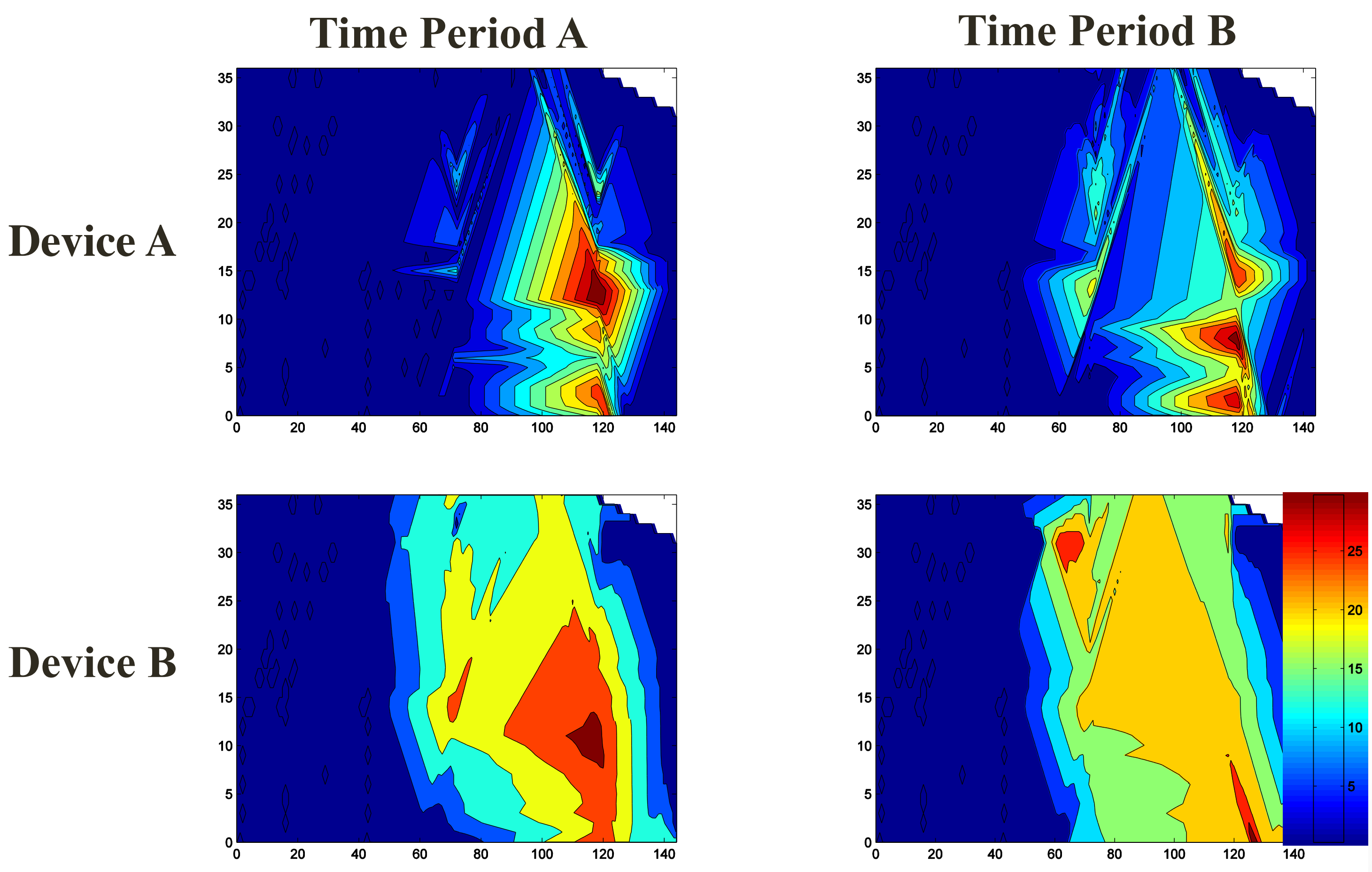
Feature-based Approaches

Encode application-specific knowledge

- For instance, sentiment analysis
  - Structural Correspondence Learning (SCL) [Biltzer *etal.* EMNLP-06]
  - Spectral Feature Alignment (SFA) [Pan *etal*. WWW-10]

Feature-based Approaches

Develop general approaches



Feature-based Approaches

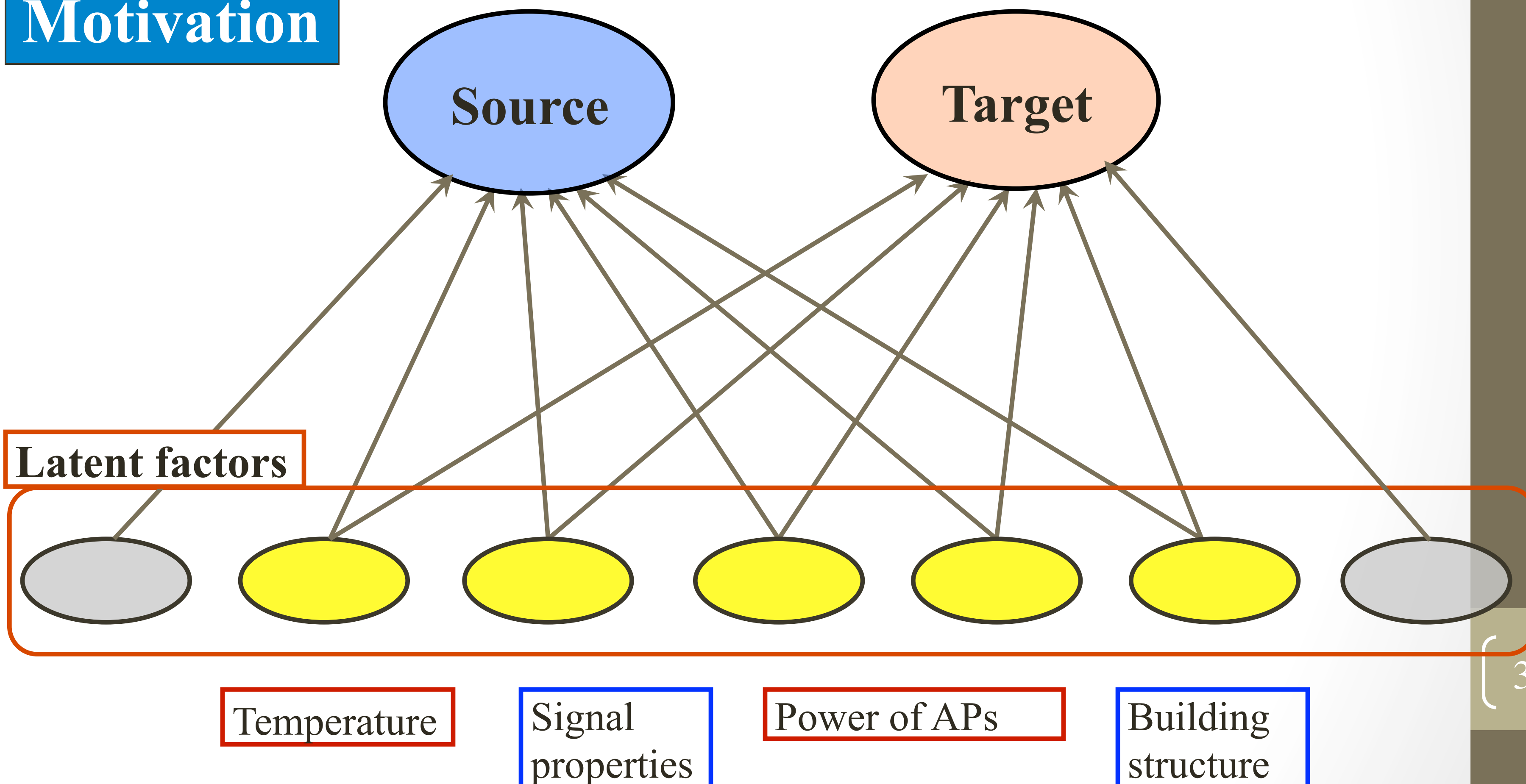
General approaches

- Learning features by minimizing distance between distributions in a latent space
- Learning features inspired by multi-task learning
- Learning features via self-taught learning

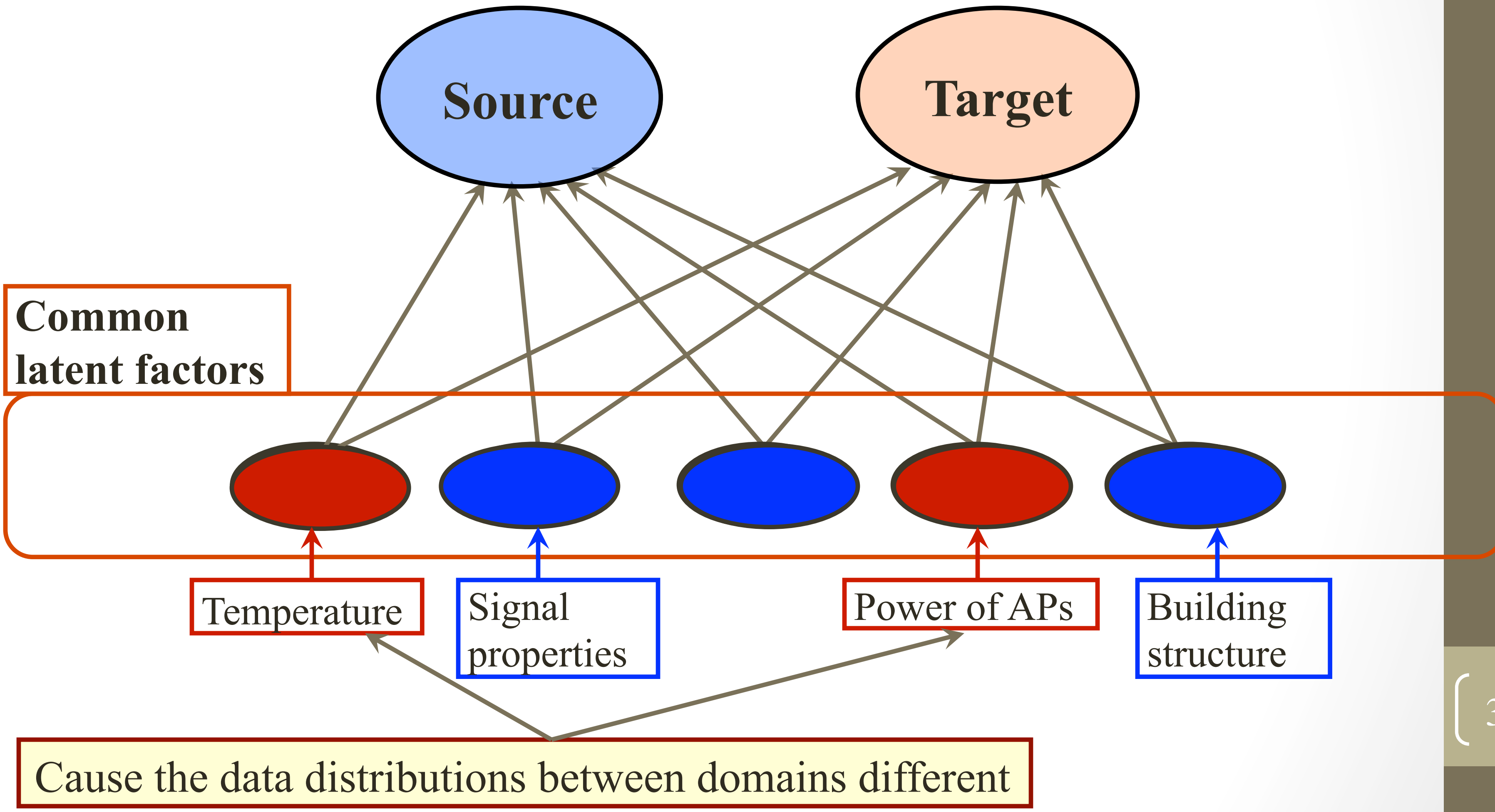
Learning Features by Minimizing Distance Between Distributions in A Latent Space

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

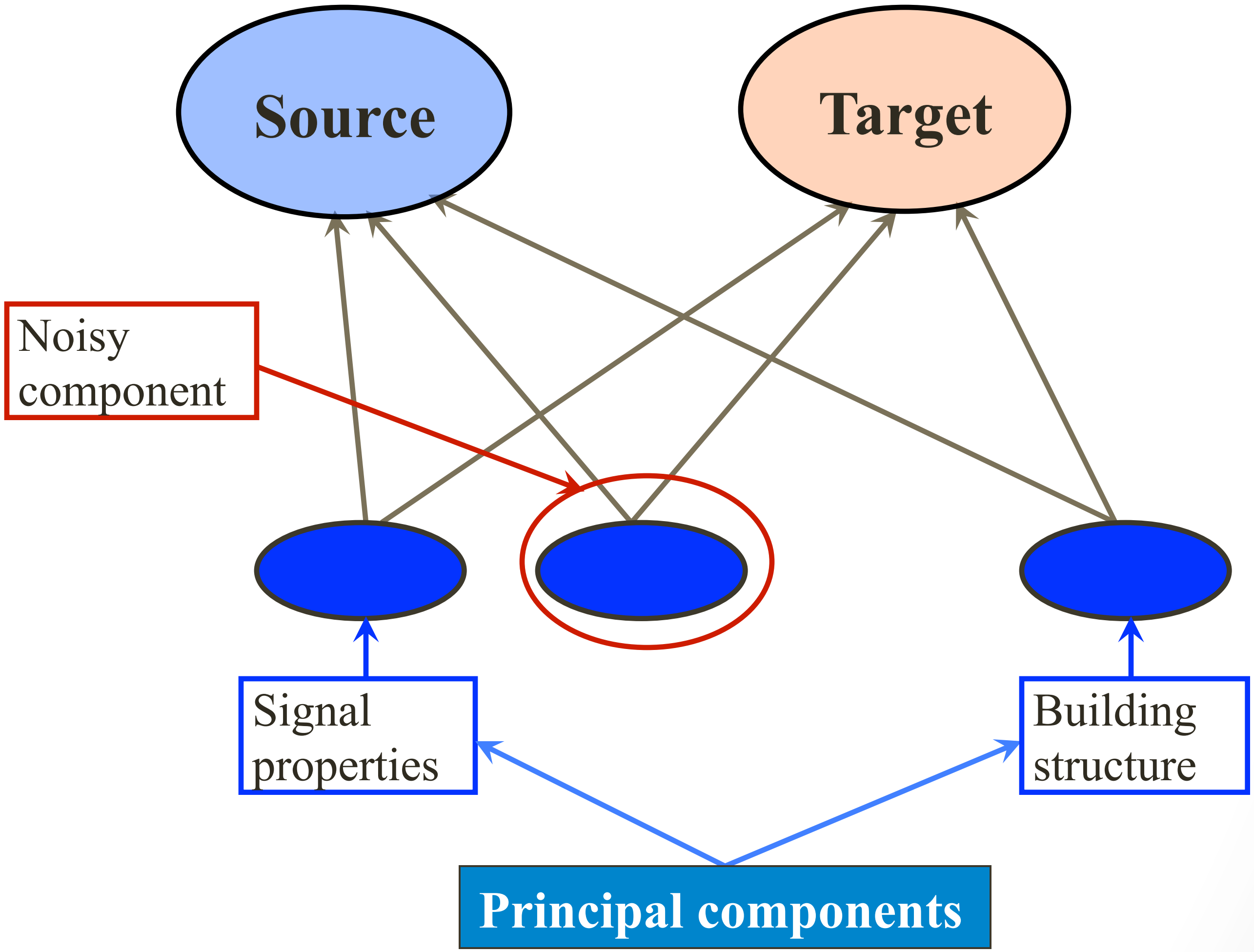
Motivation



Transfer Component Analysis (cont.)

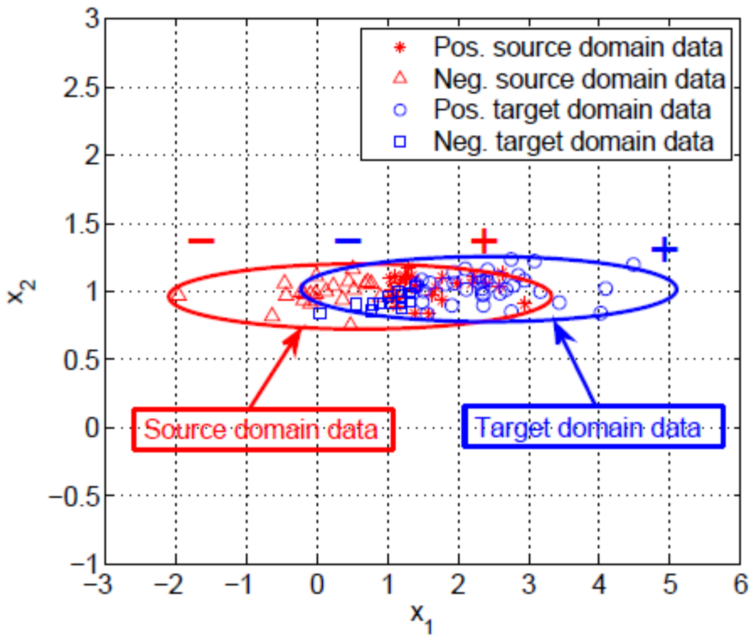


Transfer Component Analysis (cont.)



Transfer Component Analysis (cont.)

Learning φ by only minimizing distance between distributions may map the data onto noisy factors.



Transfer Component Analysis (cont.)

Main idea: the learned φ should map the source and target domain data to the latent space spanned by the factors which can reduce domain distance as well as preserving original data structure.

High level optimization problem

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

Transfer Component Analysis (cont.)

MMD

$$\begin{aligned}\mathrm{Dist}(\varphi(\mathbf{X}_S), \varphi(\mathbf{X}_T)) &= \left\| \mathbb{E}_{x\sim P_T(x)}[\Phi(\varphi(x))] - \mathbb{E}_{x\sim P_S(x)}[\Phi(\varphi(x))] \right\| \\ &\approx \left\| \frac{1}{n_S}\sum_{i=1}^{n_S} \Phi(\varphi(x_{S_i})) - \frac{1}{n_T}\sum_{i=1}^{n_T} \Phi(\varphi(x_{T_i})) \right\|\end{aligned}$$

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

$$\mathrm{Dist}(\varphi(\mathbf{X}_S), \varphi(\mathbf{X}_T)) = \mathrm{tr}(KL)$$

$$K = \begin{bmatrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{bmatrix} \in \mathbb{R}^{(n_S+n_T)\times(n_S+n_T)}, L_{ij} = \begin{cases} \frac{1}{n_S^2} & x_i, x_j \in X_S, \\ \frac{1}{n_T^2} & x_i, x_j \in X_T, \\ -\frac{1}{n_S n_T} & \text{otherwise.} \end{cases}$$

Transfer Component Analysis (cont.)

Learning $\varphi \Rightarrow$ (1) learning K [Pan *etal*., AAAI-08]

(2) low-dimensional reconstructions of $\mathbf{X}_S$ and $\mathbf{X}_T$ based on K

To minimize the distance between domains

To maximize the data variance

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

To preserve the local geometric structure

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

$$K\mathbf{1} = \mathbf{0}, \ K \succeq 0.$$

Low-dimensional constructions of $\mathbf{X}_S, \ \mathbf{X}_T \Rightarrow$ PCA on K

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

Transfer Component Analysis (cont.)

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

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

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

Resultant parametric kernel

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

Learning $\varphi \ \Rightarrow$ learning a low-rank matrix W

To minimize the distance between domains

$$\min_W \ \mathrm{tr}(W^\top KLKW) + \lambda \mathrm{tr}(W^\top W)$$

Regularization

$$\text{s.t.} \quad W^\top KHKW = I.$$

To maximize the data variance

Feature Space: Document-word co-occurrence

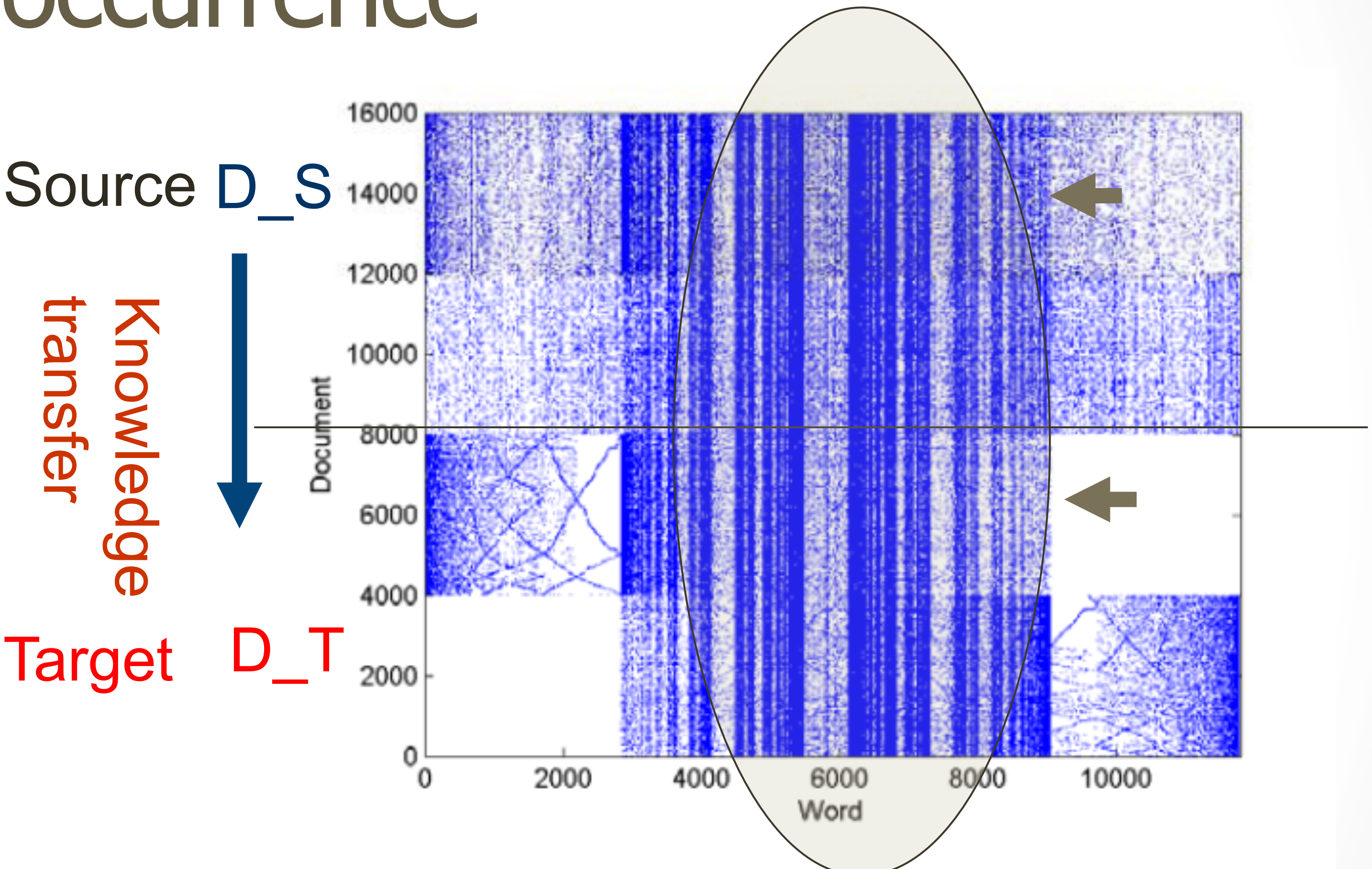
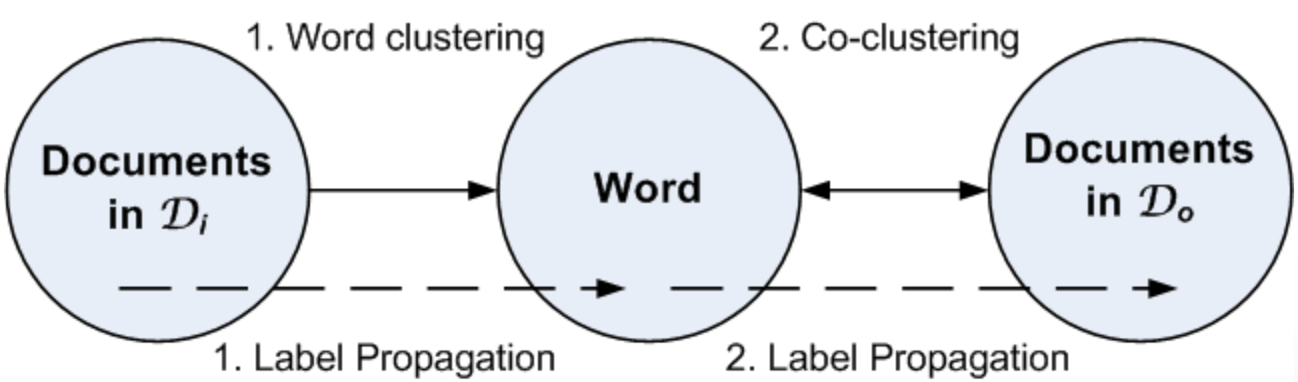


Figure 2: Document-word co-occurrence distribution on the auto vs aviation data set

Co-Clustering based Classification (KDD 2007)

- Co-clustering is applied between features (words) and target-domain documents
- Word clustering is constrained by the labels of in-domain (Old) documents
  - The word clustering part in both domains serve as a ***bridge***



Structural Correspondence Learning [Blitzer et al. ACL 2007]

- SCL: [Ando and Zhang, JMLR 2005]
- Method
  - Define pivot features: common in two domains (not buy)
  - Find non-pivot features in each domain (repetitive)
  - Build classifiers through the non-pivot Features

(1) The book is so **repetitive** that I found myself yelling …. I will definitely **not buy** another.

(2) Do **not buy** the Shark portable steamer …. Trigger mechanism is **defective**.

Book Domain ⟷ Kitchen Domain

SCL

[Blitzer et al. EMNL-06, Blitzer et al. ACL-07, Ando and Zhang JMLR-05]

Input: labeled source data $\{(\mathbf{x}_t, y_t)_{t=1}^T\}$, unlabeled data from both domains $\{\mathbf{x}_j\}$

Output: predictor $f : X \to Y$

1. Choose m pivot features. Create m binary prediction problems, $p_\ell(\mathbf{x})$, $\ell = 1 \dots m$

2. For $\ell = 1$ to m

   $$\hat{\mathbf{w}}_\ell = \underset{\mathbf{w}}{\text{argmin}} \left(\sum_j L(\mathbf{w} \cdot \mathbf{x}_j, p_\ell(\mathbf{x}_j)) + \lambda ||\mathbf{w}||^2 \right)$$

   end

3. $W = [\hat{\mathbf{w}}_1 | \dots | \hat{\mathbf{w}}_m]$, $\quad [U\ D\ V^T] = \text{SVD}(W)$, $\theta = U^T_{[1:h,:]}$

4. Return f, a predictor trained on $\left\{ \left(\begin{bmatrix} \mathbf{x}_t \\ \theta \mathbf{x}_i \end{bmatrix}, y_t \right)_{t=1}^T \right\}$

a) Heuristically choose m pivot features, which is task specific.

b) Transform each vector of pivot feature to a vector of binary values and then create corresponding prediction problem.

Learn parameters of each prediction problem

Do Eigen Decomposition on the matrix of parameters and learn the linear mapping function.

Use the learnt mapping function to construct new features and train classifiers onto the new representations.

Courtesy of Sinno Pan

Feature-based Approaches
Self-taught Feature Learning

- **Intuition:** There exist some ***high-level*** features that can help the target learning task even only a few labeled data are given
- **How to learn high-level features**
  - Sparse coding [Raina etal., 2007]
  - Deep learning [Glorot *etal.*, 2011]

Parameter-based Transfer Learning Approaches

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

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

Common part

Specific part for individual task

$$\theta_S = \theta_0 + v_S$$

$$\theta_T = \theta_0 + v_T$$

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

Parameter-based Approaches

Multi-task Parameter Learning (cont.)

A general framework:

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

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

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

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

[Zhang and Yeung, UAI-10]

$$f(\Theta) = \mathrm{tr}(\Theta^\top \Sigma^{-1} \Theta)$$

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

[Agarwal *etal*, NIPS-10]

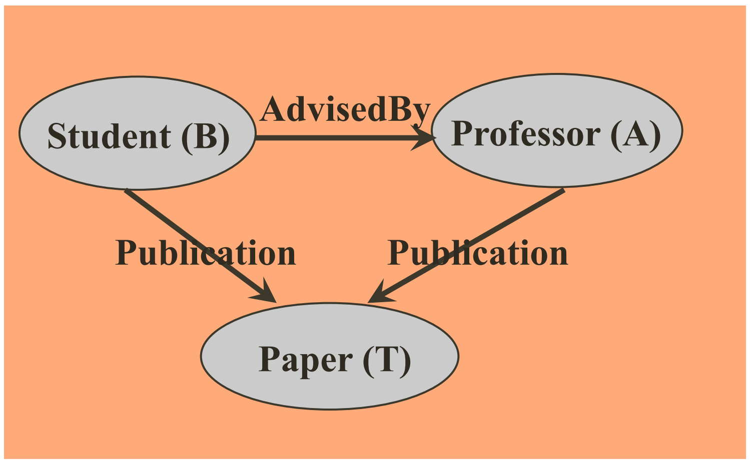
$$f(\Theta) = \sum_{t \in \{S,T\}} \left\| \theta_t - \widetilde{\theta}_t^{\mathcal{M}} \right\|^2$$

Relational Transfer Learning Approaches

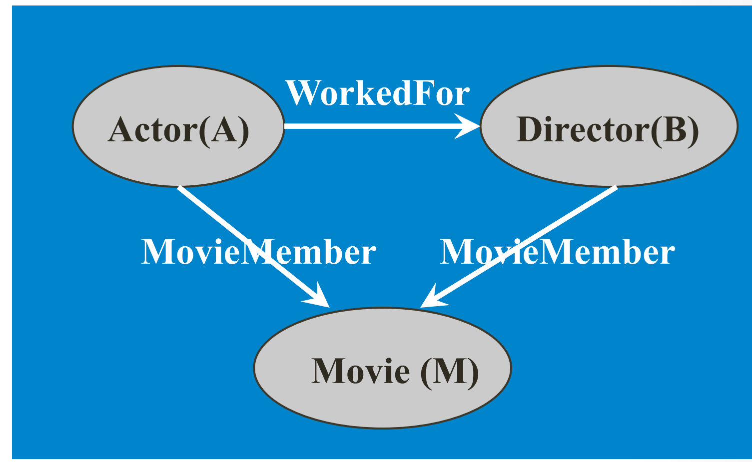
- **Motivation:** If two relational domains (data is non-i.i.d) are related, they may share some similar relations among objects. These relations can be used for knowledge transfer across domains.

Relational Transfer Learning Approaches (cont.)

Academic domain (source)



Movie domain (target)



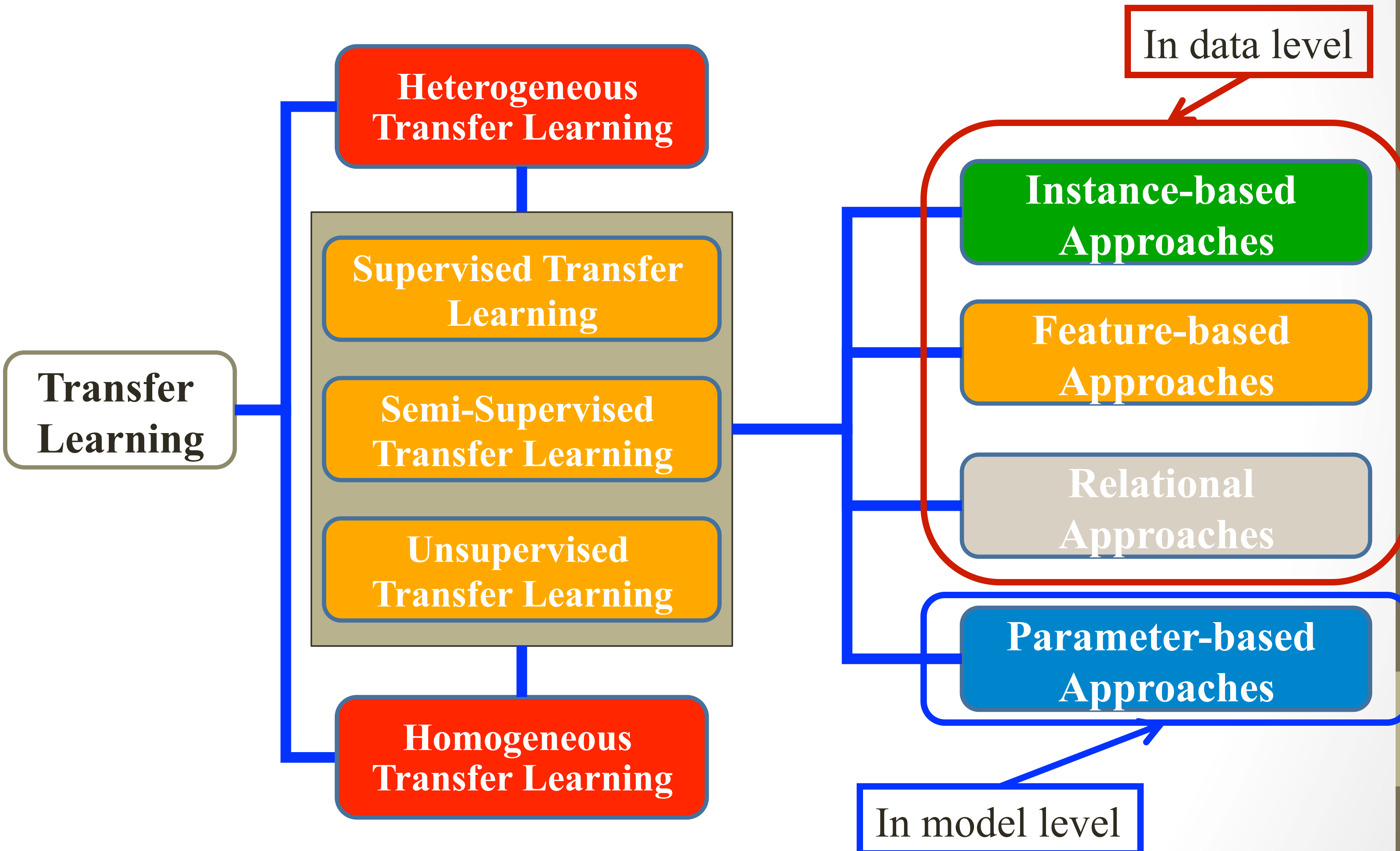
AdvisedBy (B, A) ∧ Publication (B, T)
=> Publication (A, T)

WorkedFor (A, B) ∧ MovieMember (A, M)
=> MovieMember (B, M)

P1(x, y) ∧ P2 (x, z) => P2 (y, z)

[Davis and Domingos, ICML-09]

Summary



In data level
Heterogeneous Transfer Learning
Supervised Transfer Learning
Semi-Supervised Transfer Learning
Unsupervised Transfer Learning
Transfer Learning
Homogeneous Transfer Learning
Instance-based Approaches
Feature-based Approaches
Relational Approaches
Parameter-based Approaches
In model level

Some Research Issues in Transfer Learning

- When should transfer learning be applied
- Transfer learning across heterogeneous feature spaces or different label spaces
- Active learning meets transfer learning
- Transfer learning meets lifelong learning
- Transfer learning to novel application areas

Reference

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

Reference (cont.)

- [Pan *etal.*, Domain Adaptation via Transfer Component Analysis, IJCAI 2009, TNN 2011]
- [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 *etal*, Learning Multiple Tasks using Manifold Regularization, NIPS 2010]
- [Argyriou *etal.*, Multi-Task Feature Learning, NIPS 2007]
- [Ando and Zhang, A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data, JMLR 2005]
- [Ji *etal*, Extracting Shared Subspace for Multi-label Classification, KDD 2008]
- [Dai *etal.,* Boosting for Transfer Learning, ICML 2007]

Reference (cont.)

- [Raina *etal.*, Self-taught Learning: Transfer Learning from Unlabeled Data, ICML 2007]
- [Glorot *etal.*, 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]
- [Sugiyama *etal.*, Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation, NIPS 2007]
- [Kanamori *etal.*, A Least-squares Approach to Direct Importance Estimation, JMLR 2009]
- [Huang *etal.*, Correcting Sample Selection Bias by Unlabeled Data, NIPS 2006]
- [Zadrozny, Learning and Evaluating Classifiers under Sample Selection Bias, ICML 2004]

Selected Applications of Transfer Learning

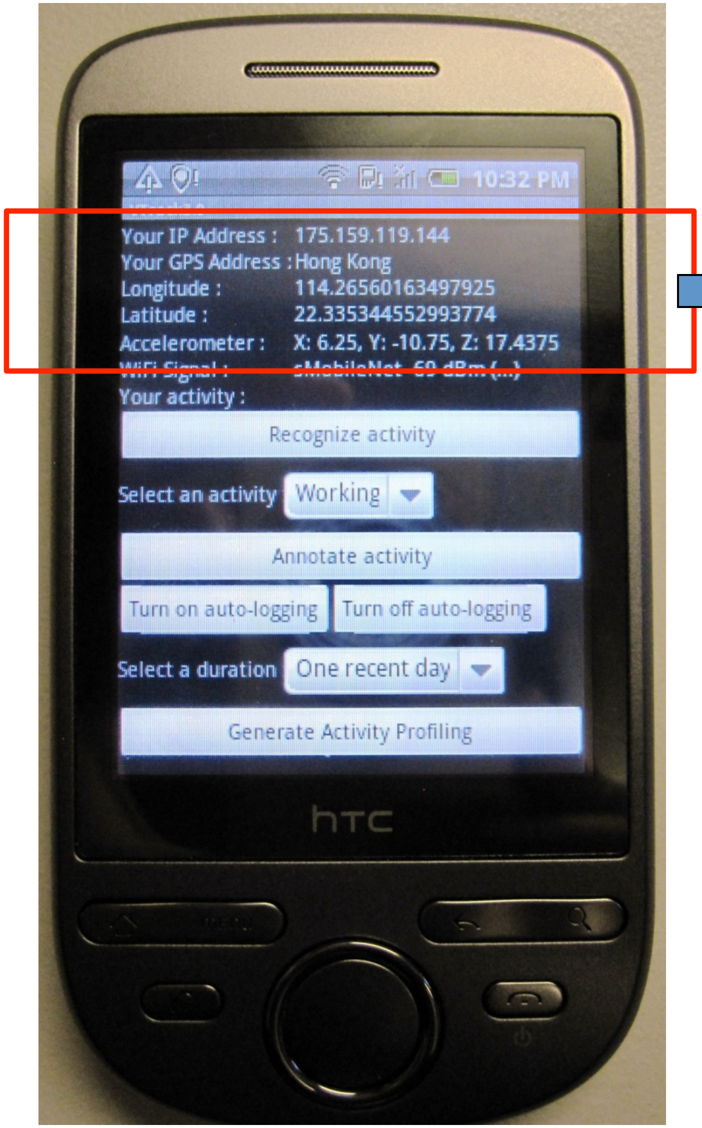
Qiang Yang, Huawei Noah's Ark Lab
and HKUST

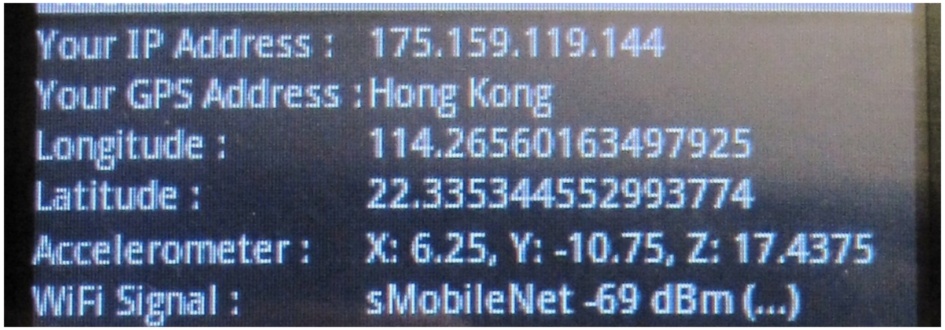
http://www.noahlab.com.hk

Part 1. Cross Domain Transfer Learning for Activity Recognition

- Vincent W. Zheng, Derek H. Hu and Qiang Yang. Cross-Domain Activity Recognition. In *Proceedings of the 11th International Conference on Ubiquitous Computing* (**Ubicomp-09**), Orlando, Florida, USA, Sept.30-Oct. 3, 2009.
- Derek Hao Hu, Qiang Yang. Transfer Learning for Activity Recognition via Sensor Mapping. *In Proceedings of the 22nd International Joint Conference on Artificial Intelligence* (IJCAI-11), Barcelona, Spain, July 2011

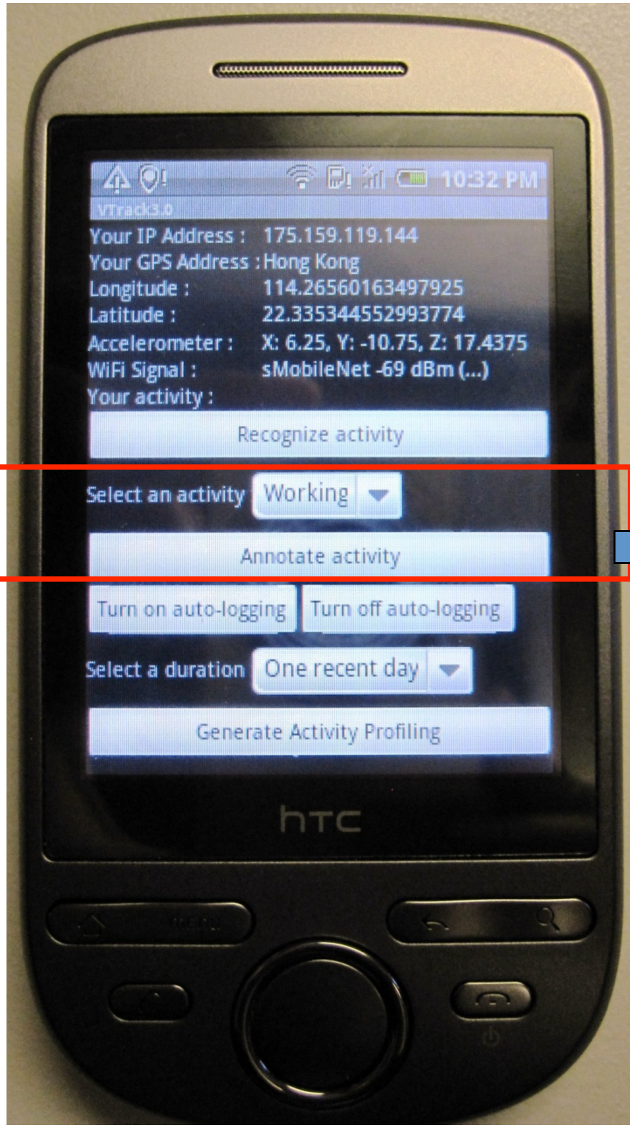
eHealth Demo

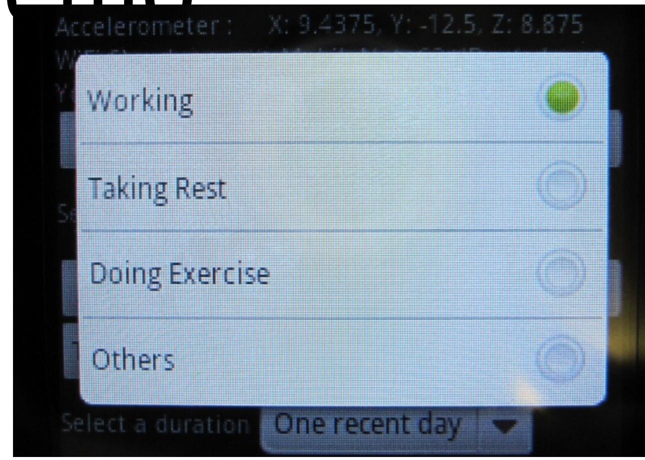


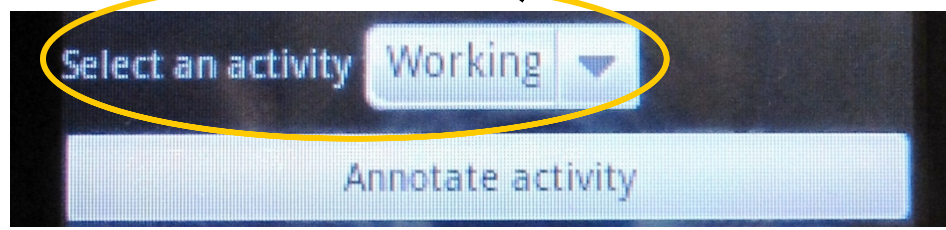


Sensor data

eHealth demo

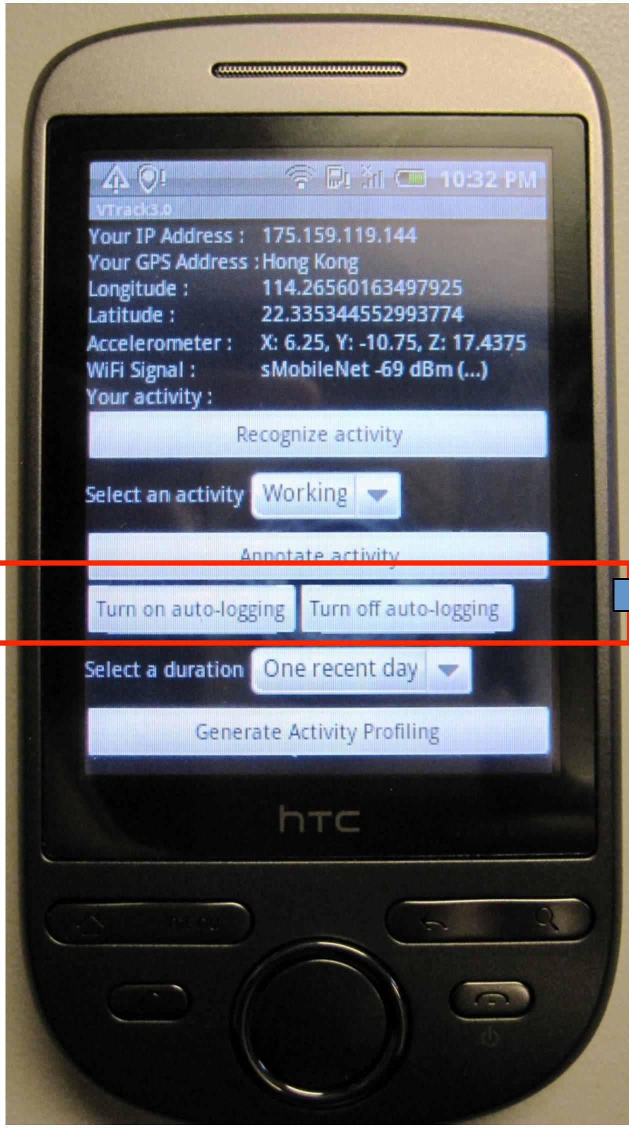


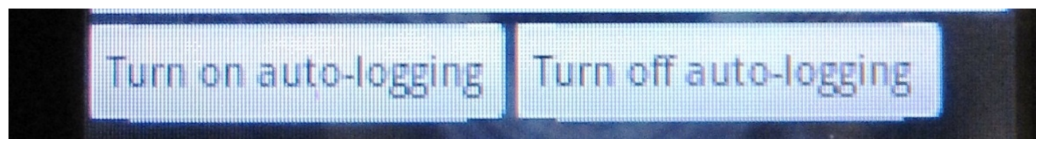




Activity annotation

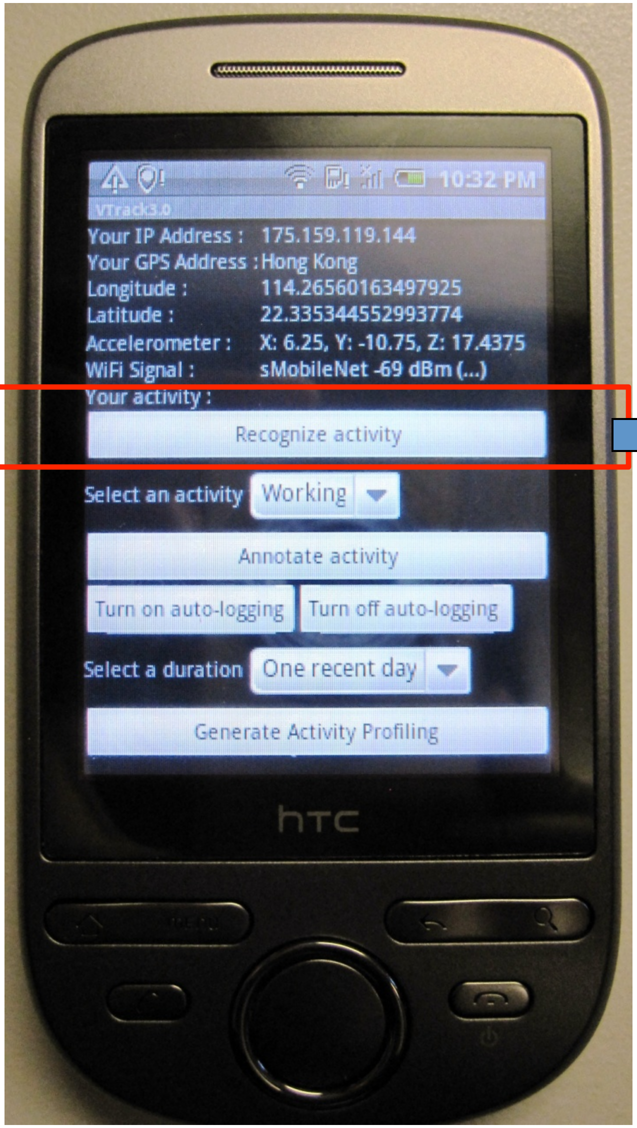
eHealth demo

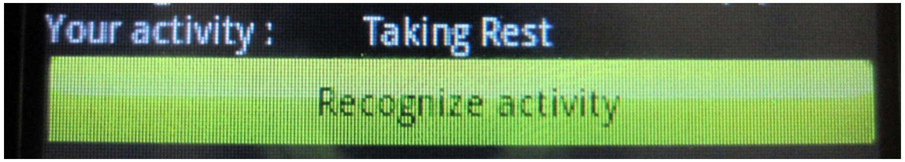




Auto logging / activity recognition
(service in background)

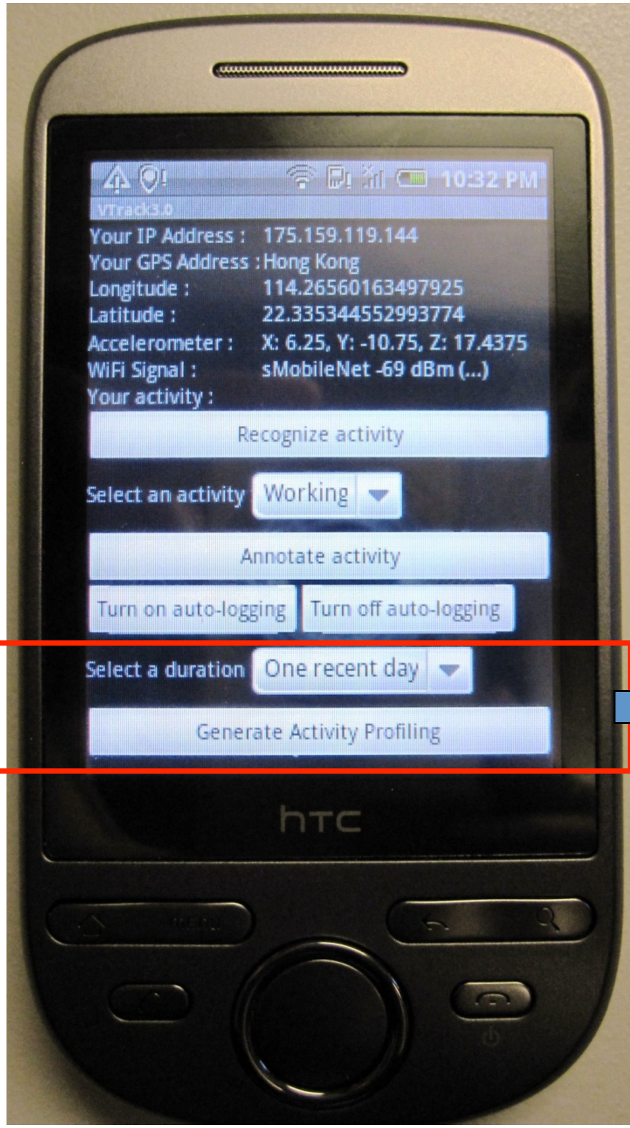
eHealth demo

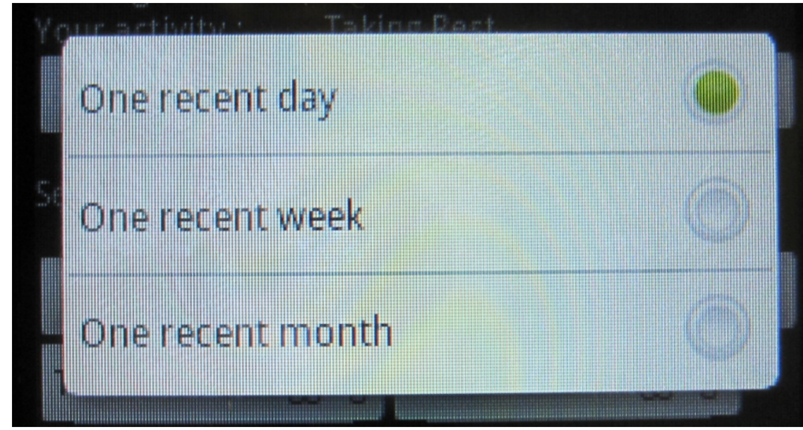


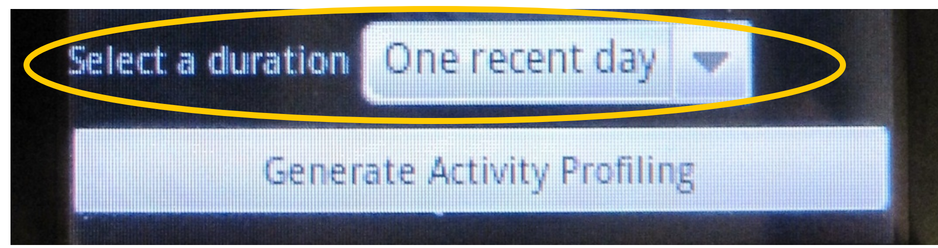


Real-time activity recognition

eHealth demo

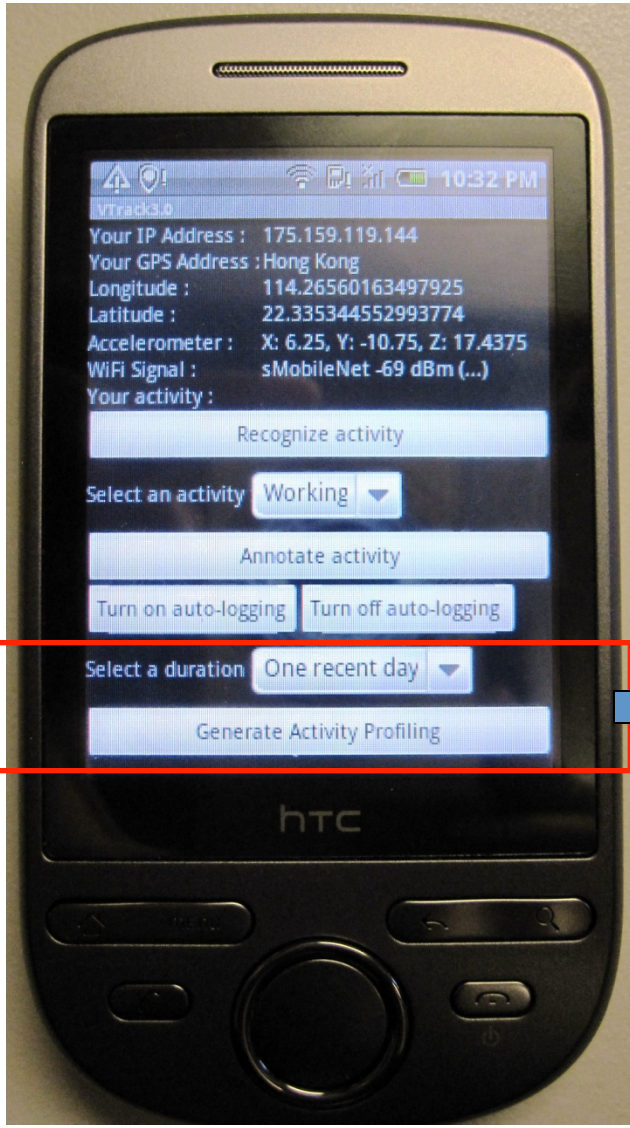


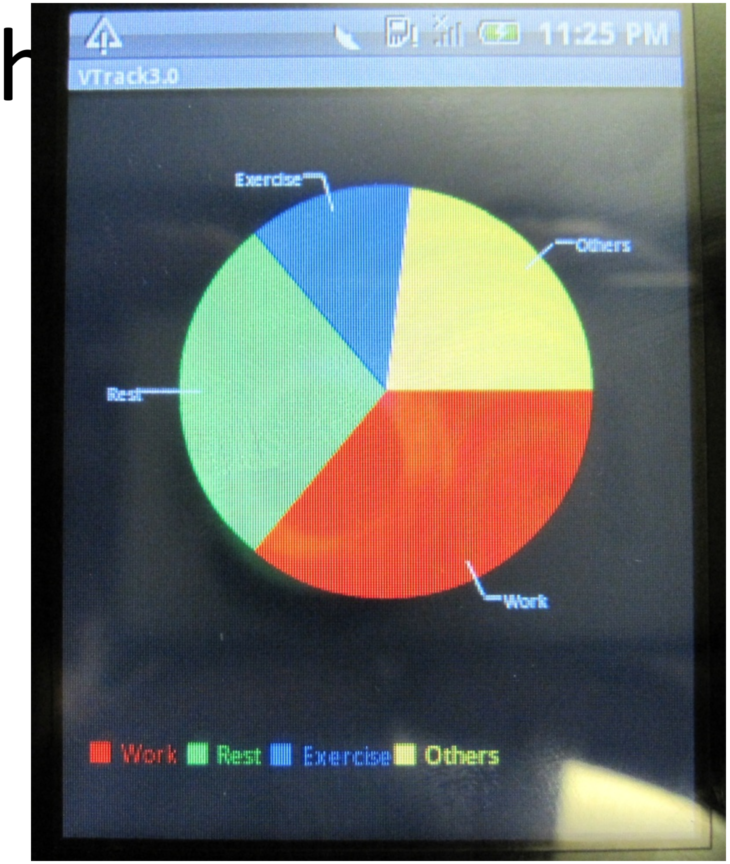


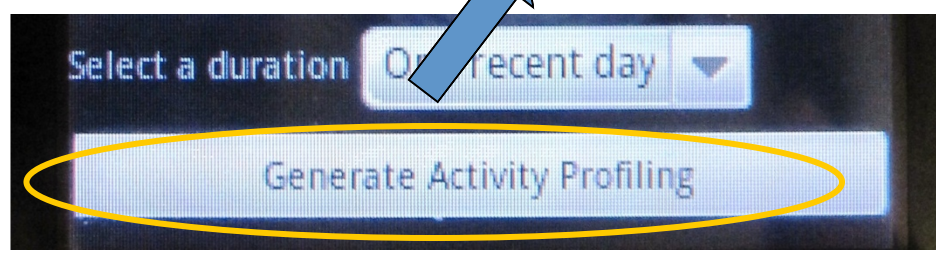


Activity profiling

eHealth

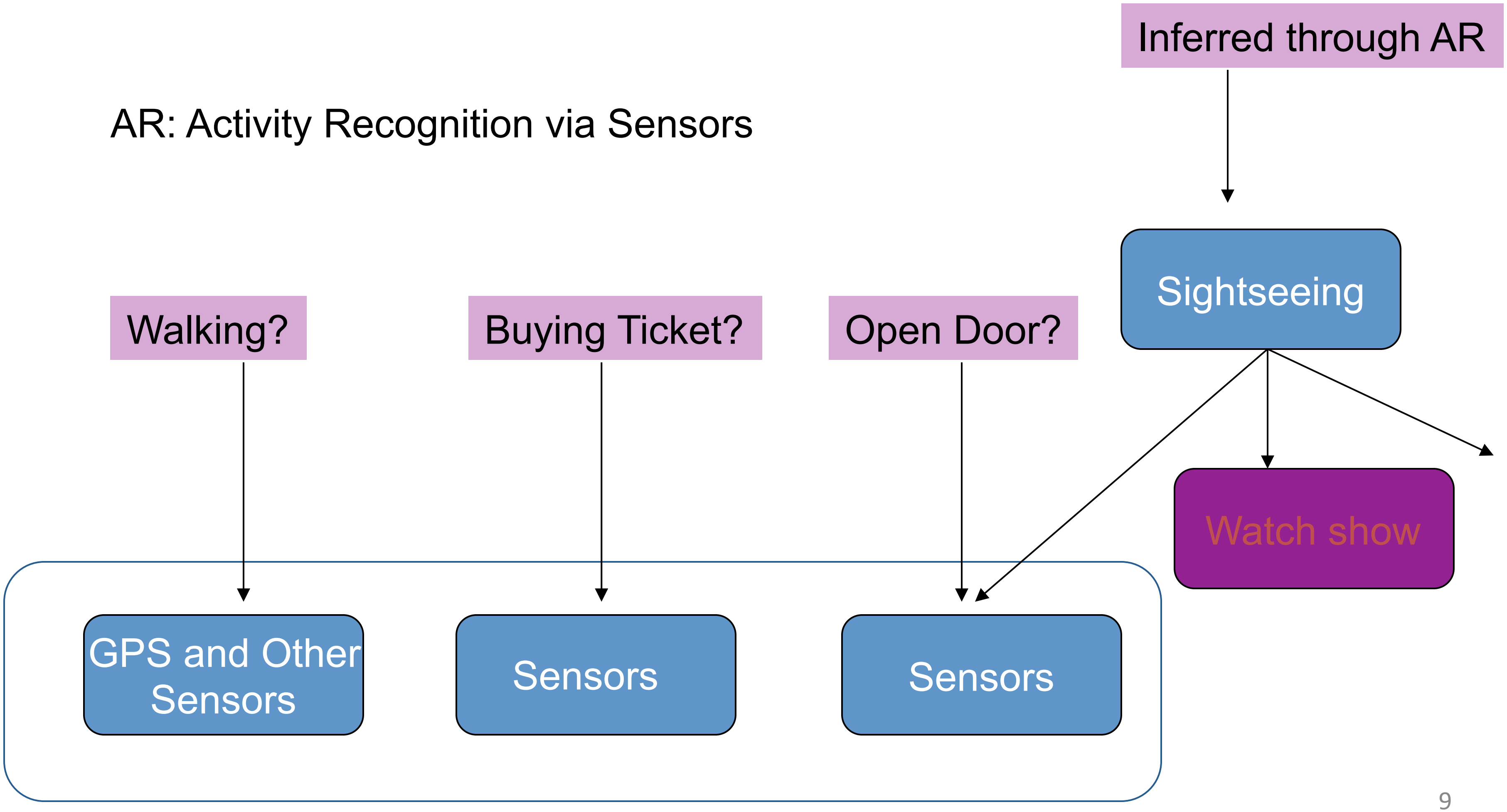






Activity profiling for health management

Key Problem: Recognizing Actions and Context (Locations)



1. Cross-Domain Activity Recognition
[Zheng, Hu, Yang: UbiComp-2009, PCM-2011]

- Challenge:
  – Some activities without data (partially labeled)
- Cross-domain activity recognition
  – Use other activities with available labeled data

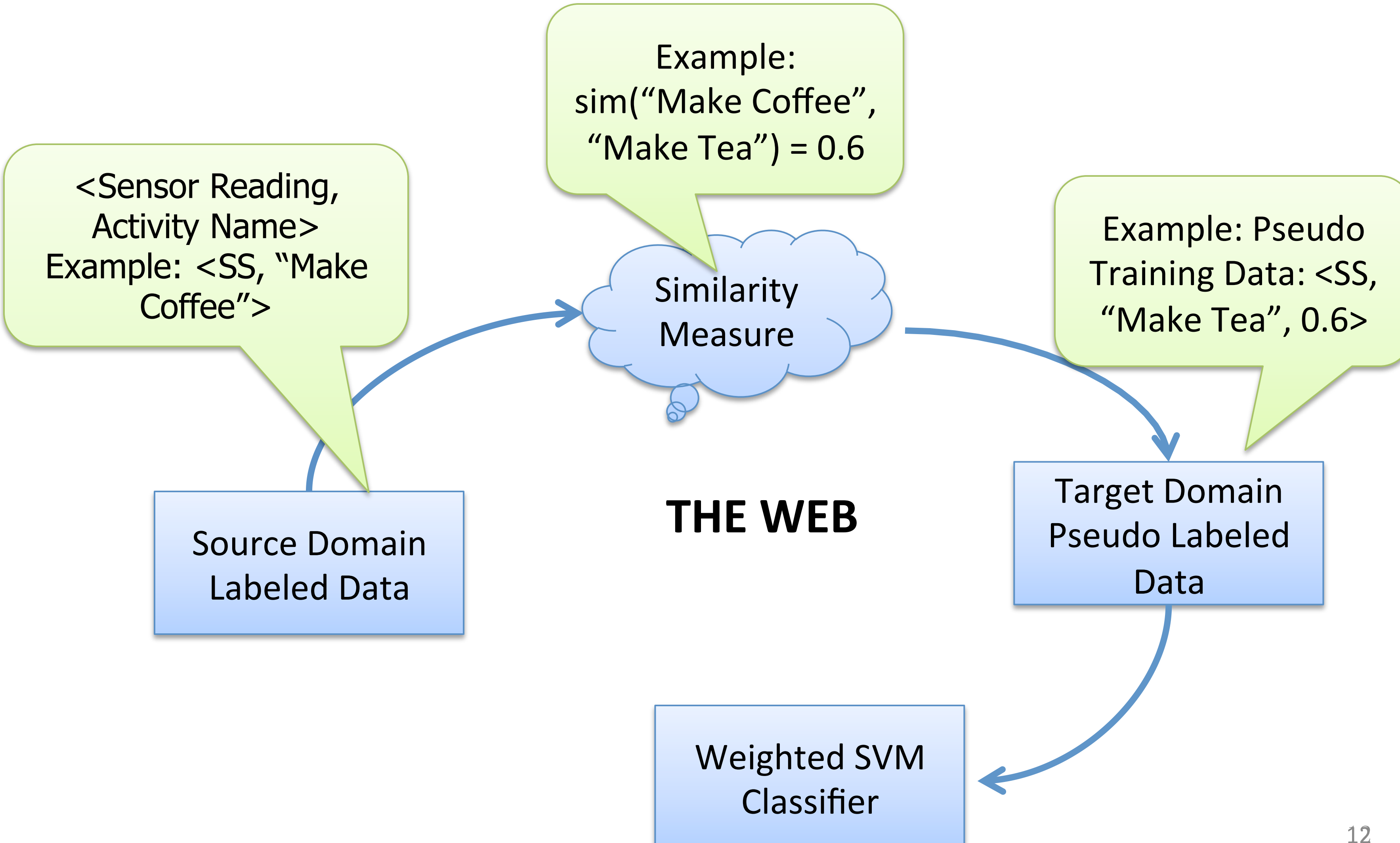
Making coffee

- Happen in kitchen
- Use cup, pot
- …

Making tea

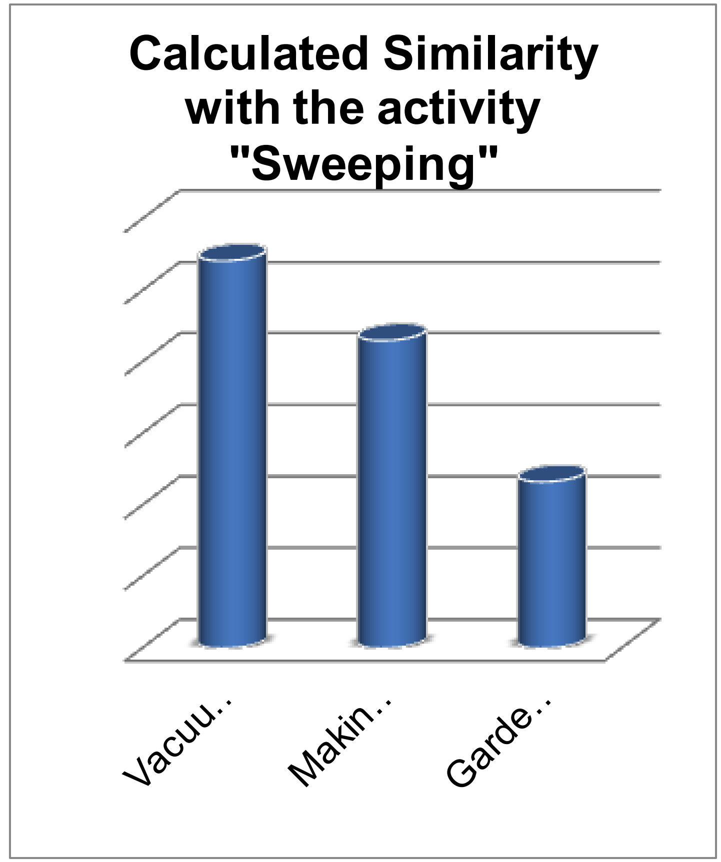
Source Domain
Cleaning Indoor
Sweeping
Swiftering
Mopping
Vacuuming
Dusting
Sweeping
Making-the-bed
Putting-things-away
Organizing
Disposing-Garbage
Taking-out-trash
Dealing-with-Garbage
Cleaning-a-surface
Scrubbing
Cleaning-a-surface
Cleaning miscellaneous
Cleaning-miscellaneous
Cleaning-background
Cleaning-background
Cleaning Indoor
Activity Transfer
Target Domain 1
Gardening
Mowing lawn
Gardening
Yardwork-miscellaneous
Yardwork-miscellaneous
Yardwork
Laundry
Washing-laundry
Drying-laundry
Washing/Drying-laundry
Washing-laundry-background
Drying-laundry-background
Washing/Drying-laundry-background
Folding-laundry
Putting-away-laundry
Ironing
Dealing-with-clothes
Laundry-miscellaneous
Laundry-miscellaneous
Laundry
Target Domain 2
Hand-washing-dishes
Drying-dishes
Putting-away-dishes
Dealing-with-dishes
Loading-dishwasher
Unloading-dishwasher
Loading/unloading-dishwasher
Dishwashing-miscellaneous
Dishwashing-miscellaneous
Dishwashing
Dishwashing

System Workflow



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"



Datasets: MIT PlaceLab

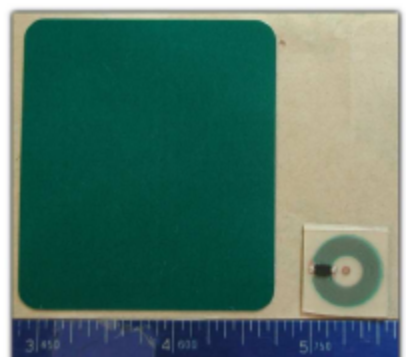
http://architecture.mit.edu/house_n/placelab.html

- 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

1	Using the bathroom
2	Making oatmeal
3	Making soft-boiled eggs
4	Preparing orange juice
5	Making coffee
6	Making tea
7	Making or answering a phone call
8	Taking out the trash
9	Setting the table
10	Eating breakfast
11	Clearing the table



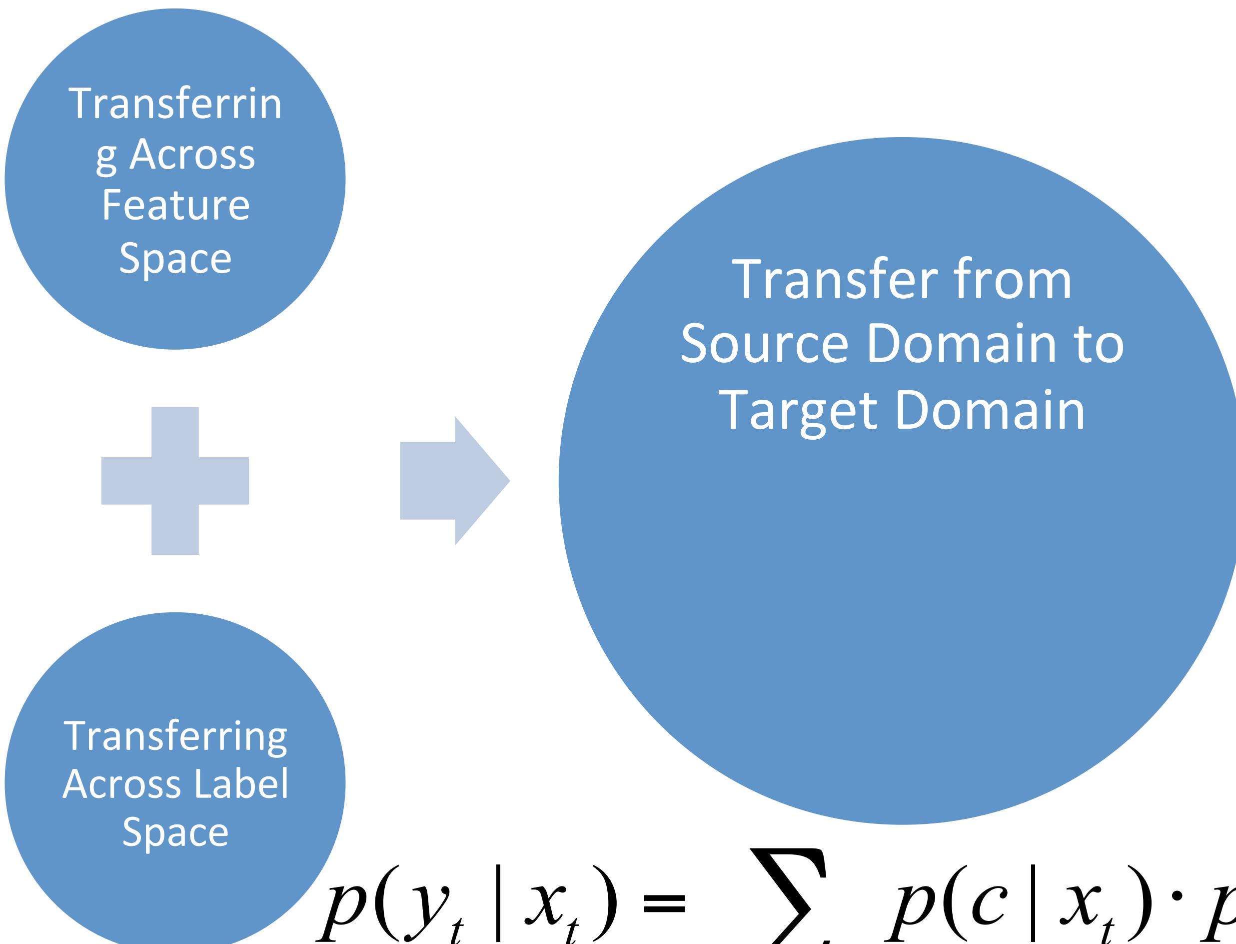
Picture excerpted from [Patterson, Fox, Kautz, Philipose, ISWC2005].

Cross-Domain AR: Performance

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

- **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



$$p(y_t \mid x_t) = \sum_{c^{(i)} \in \mathbf{L}_s} p(c \mid x_t) \cdot p(y_t \mid c)$$

Proposed Approach

- Final goal: Estimate $p(\mathbf{y}_t \mid \mathbf{x}_t)$
  - We have $p(\mathbf{y_t}|\mathbf{x_t}) = \sum_{\mathbf{c}^{(i)} \in \mathcal{L}_s} p(\mathbf{c}|\mathbf{x_t}) \cdot p(\mathbf{y_t}|\mathbf{c})$
  - $p(\mathbf{y_t}|\mathbf{x_t}) \approx \underbrace{p(\hat{\mathbf{c}}|\mathbf{x_t})}_{\text{Feature Transfer}} \cdot \underbrace{p(\mathbf{y_t}|\hat{\mathbf{c}})}_{\text{Label Transfer}} \quad (\hat{\mathbf{c}} = \arg\max_{\mathbf{c} \in \mathcal{L}_s} p(\mathbf{c}|\mathbf{x_t}))$ e:

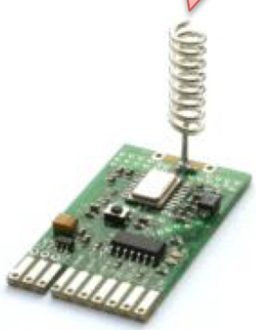
Experiments

- 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

State-based sensors for UvA dataset

A series of different wired sensors for MIT dataset

RFID sensor for Intel Research Lab Dataset



Experiments: Different Feature & Label Spaces

K	MIT $\rightarrow$ UvA Acc(Var)
K = 5	**59.8% (4.2%)**
K = 10	57.5% (4.1%)
K = 15	51.0% (4.8%)
K = 20	41.0% (4.1%)
Unsupervised	47.3%(4.1%)

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

K	MIT $\rightarrow$ Intel Acc(Var)
K = 5	60.5% (4.2%)
K = 10	**61.2% (3.8%)**
K = 15	53.2% (4.1%)
K = 20	42.0% (2.5%)
Unsupervised	42.8%(3.8%)

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

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

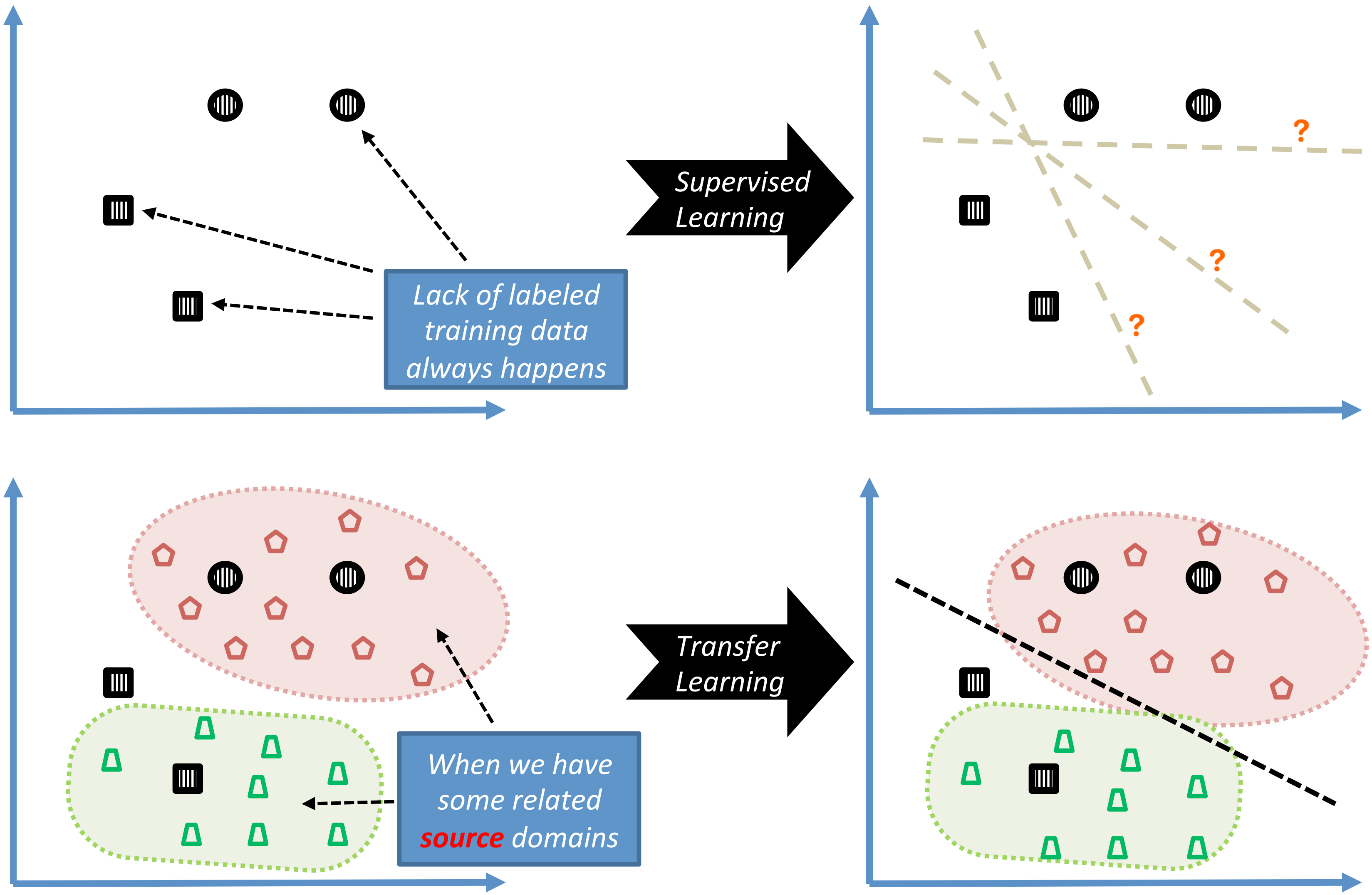
Part 2. Source-Free Transfer Learning

- Source Free Transfer Learning
- Evan Wei Xiang, Sinno Jialin Pan, Weike Pan, Jian Su and Qiang Yang. Source-Selection-Free Transfer Learning. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11), Barcelona, Spain, July 2011.

Source-Selection *free* Transfer Learning

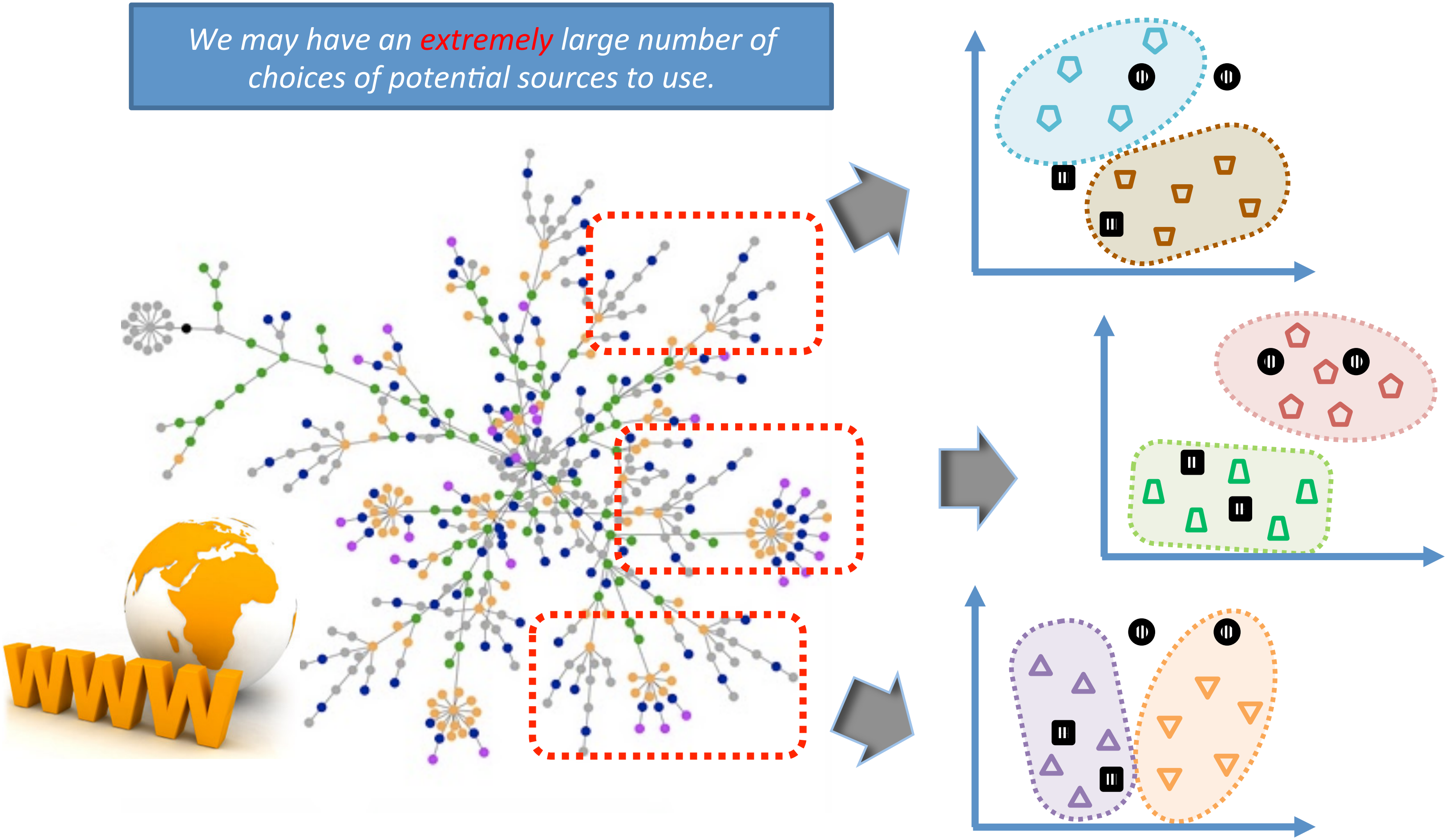
Evan Xiang, Sinno Pan, Weike Pan,
Jian Su, Qiang Yang

Transfer Learning



Where are the "right" source data?

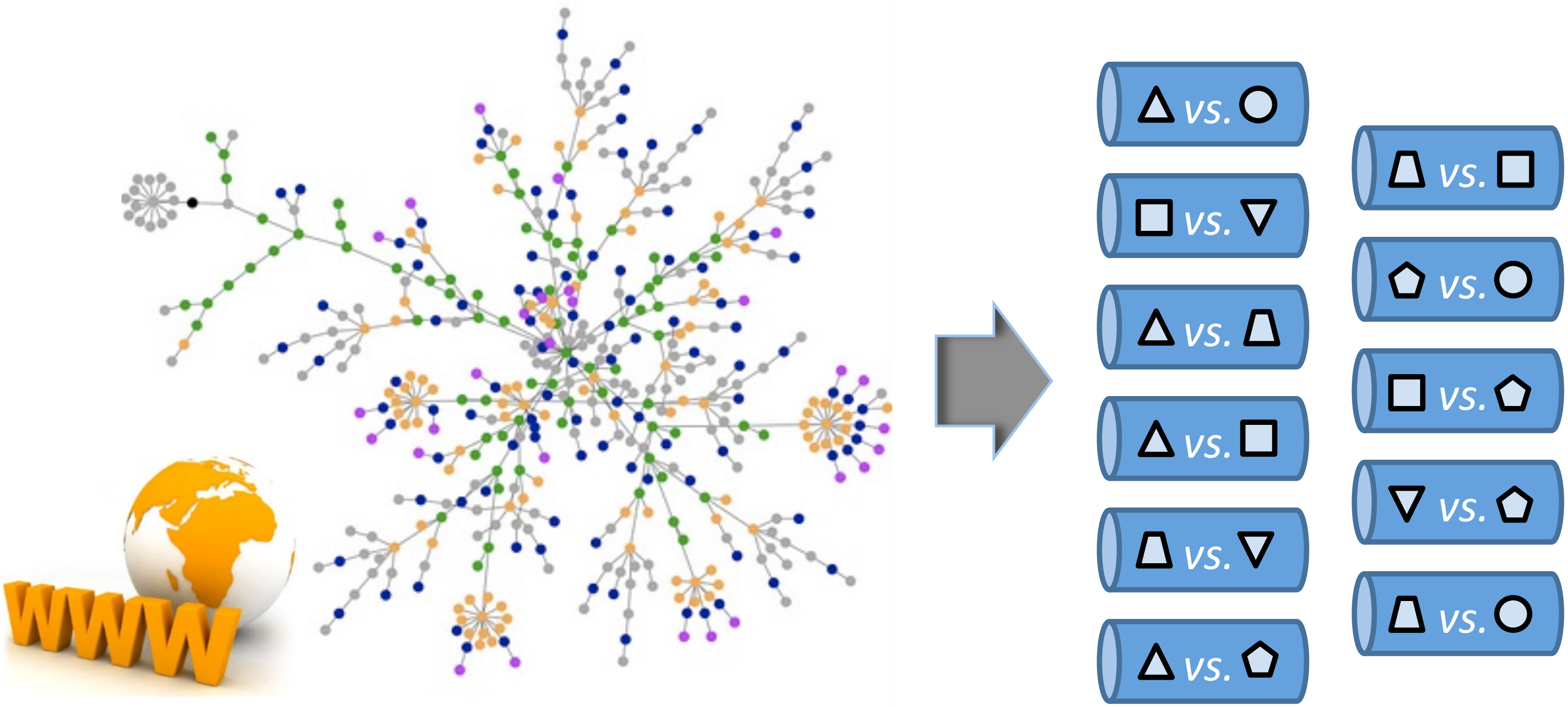
We may have an extremely large number of choices of potential sources to use.



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***

SSFTL – Building base models

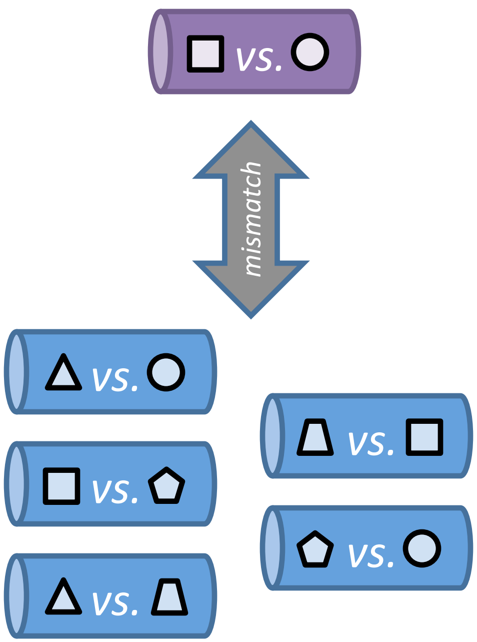


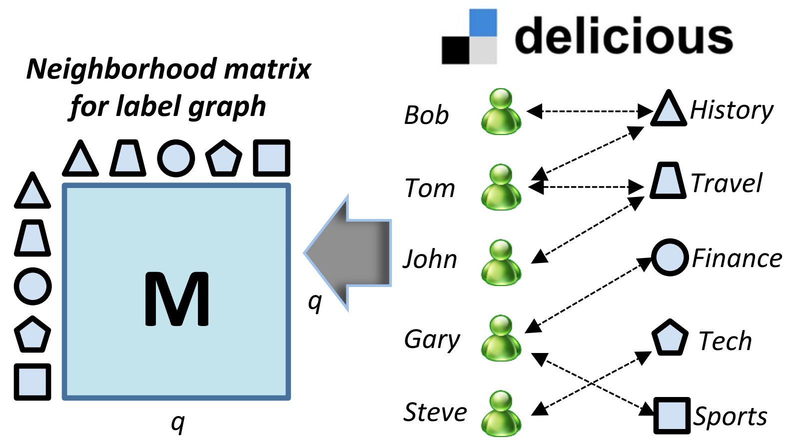
*From the taxonomy of the online information source, we can "**Compile**" a lot of base classification models*

SSFTL – Label Bridging via Laplacian Graph Embedding

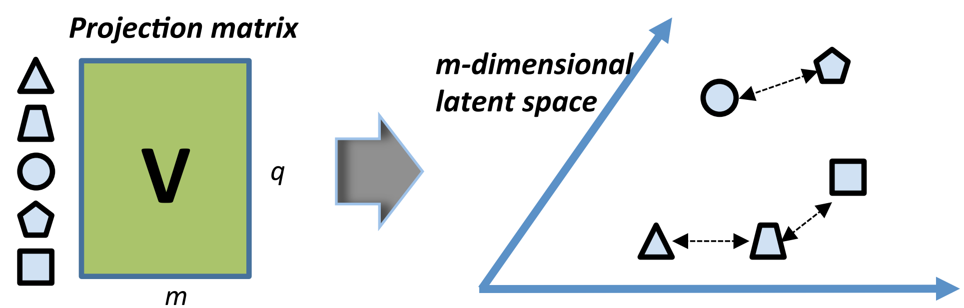
Problem

However, the ***label spaces*** *of the based classification models and the target task can be* ***different***





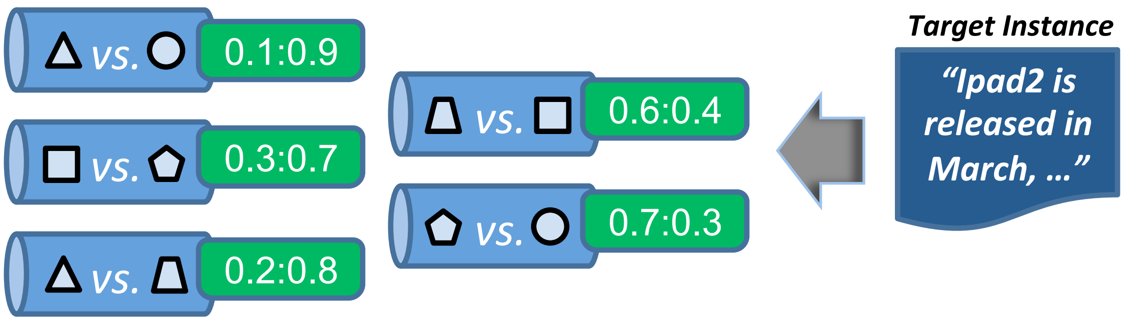
Laplacian Eigenmap
[Belkin & Niyogi,2003]



Since the label names are usually short and sparse, , in order to uncover the intrinsic relationships between the target and source labels, we turn to some ***social media*** *such as Delicious, which can help to bridge different label sets together.*

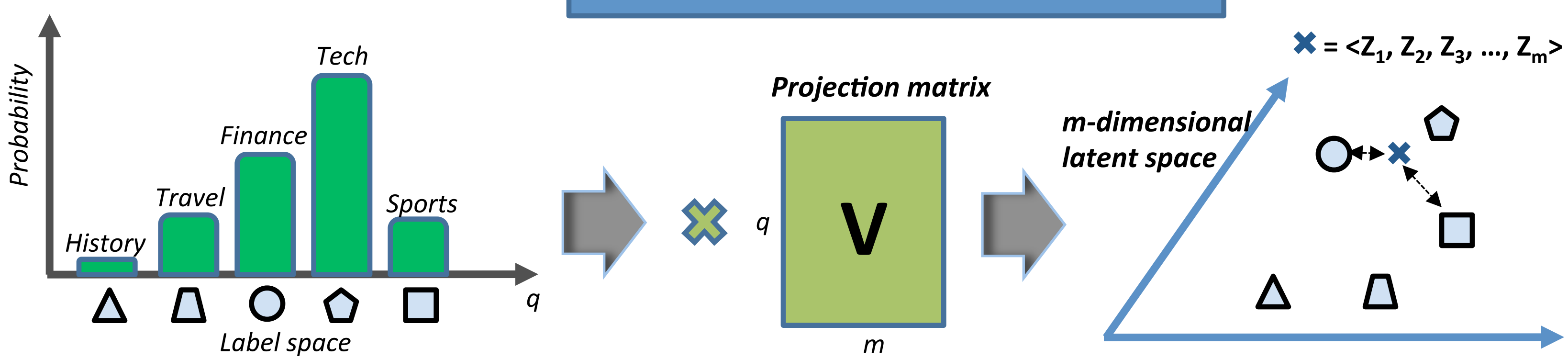
The ***relationships*** *between labels, e.g., similar or dissimilar, can be represented by the* ***distance*** *between their corresponding prototypes in the latent space, e.g., close to or far away from each other.*

SSFTL – Mapping the target instance using the base classifiers & the projection matrix V



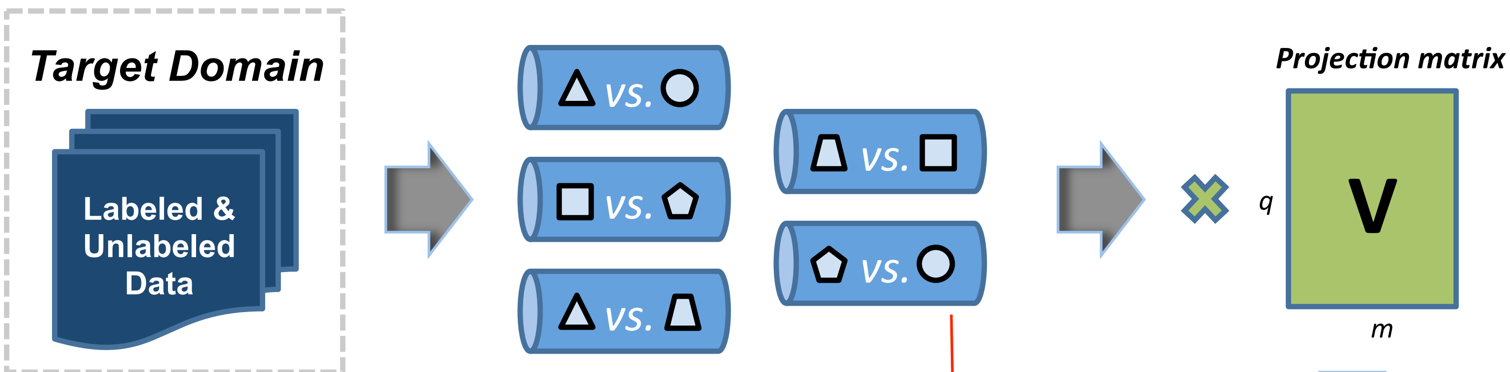
*For each target instance, we can obtain a **combined result on the label space** via aggregating the predictions from all the base classifiers*

*Then we can use the **projection matrix V** to transform such combined results from the **label** space to a **latent** space*



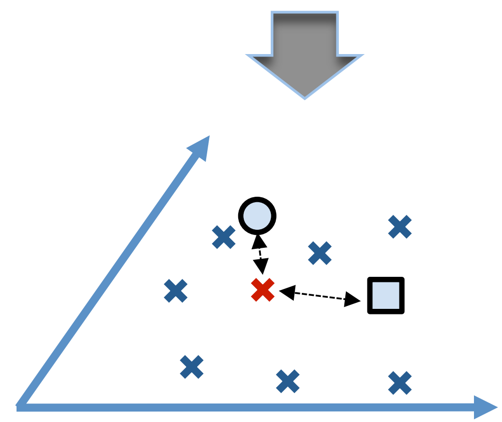
*However, do we need to recall the base classifiers during the **prediction** phase? The answer is **No**!*

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

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



Loss on unlabeled data

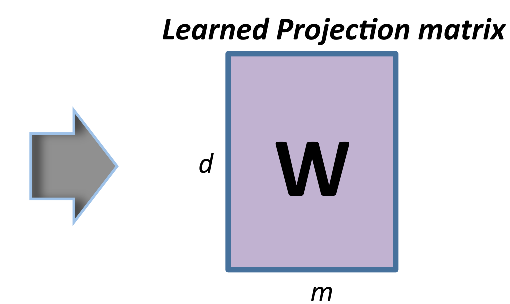
$$\Omega_{\mathbf{D}^u_T}(\mathbf{W}) = \frac{1}{n-\ell} \|\mathbf{W}'\mathbf{X}^u - \mathbf{V}'\mathbf{F}^u_S\|_F^2$$

Loss on labeled data

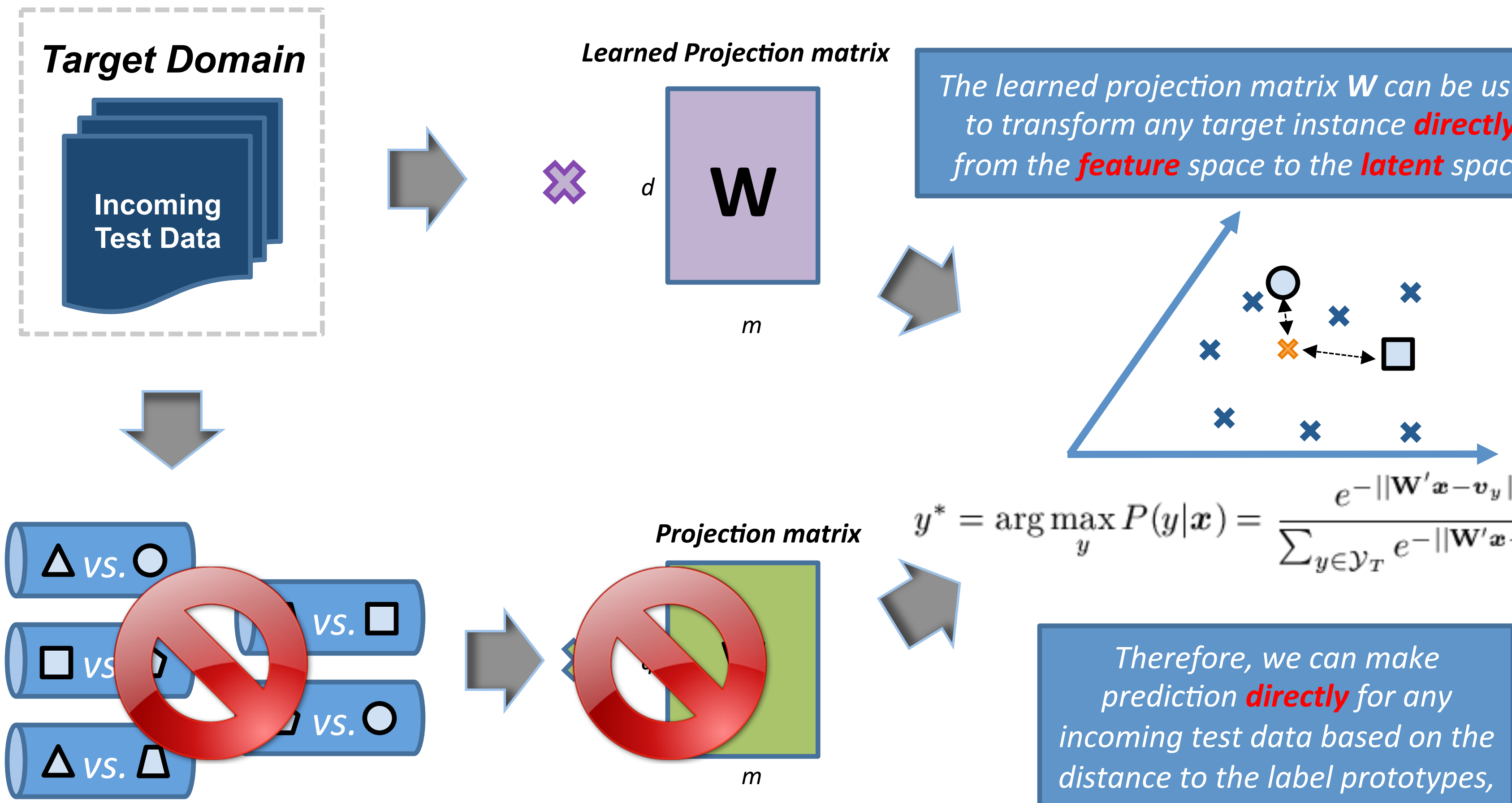
$$\Omega_{\mathcal{D}^\ell_T} = \frac{1}{\ell} \|\mathbf{W}'\mathbf{X}^\ell - \mathbf{V}'_T\phi(\mathbf{Y}^\ell))\|_F^2$$

Our regression model

$$\min_{\mathbf{W}} \Omega_{\mathcal{D}^\ell_T}(\mathbf{W}) + \lambda_1 \|\mathbf{W}\|_F^2 + \lambda_2 \Omega_{\mathcal{D}^u_T}(\mathbf{W})$$



SSFTL – Making predictions for the incoming test data



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

$$y^* = \arg\max_y P(y|\boldsymbol{x}) = \frac{e^{-\|\mathbf{W}'\boldsymbol{x}-\boldsymbol{v}_y\|_2^2}}{\sum_{y\in\mathcal{Y}_T} e^{-\|\mathbf{W}'\boldsymbol{x}-\boldsymbol{v}_y\|_2^2}}$$

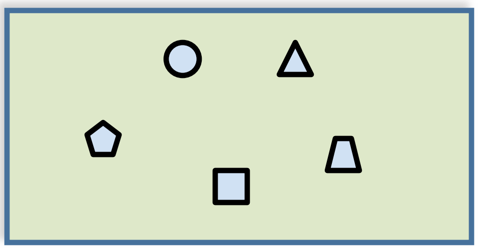
Therefore, we can make prediction **directly** for any incoming test data based on the distance to the label prototypes, ***without calling the base classification models***

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)

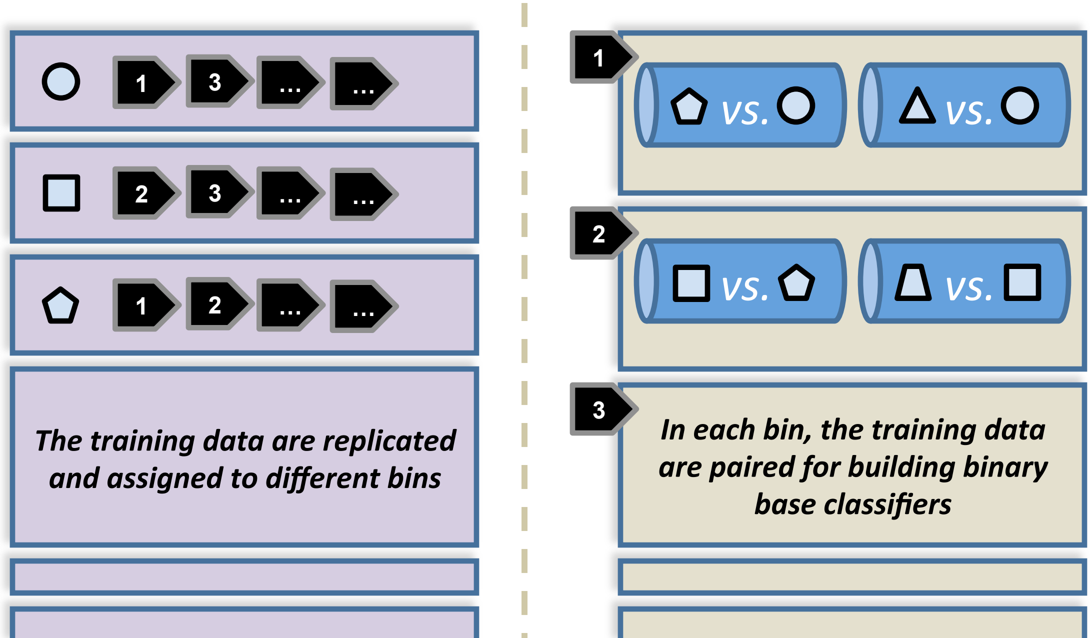
SSFTL - Building base classifiers Parallelly using MapReduce

Input



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***.

Map | **Reduce**



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

Experiments - Results

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

Dataset	0		5			10			20		
	RG	SSFTL	SVM	TSVM	SSFTL	SVM	TSVM	SSFTL	SVM	TSVM	SSFTL
20NG	50.0	**80.3**	69.8	75.7	**80.6**	72.5	81.0	**81.6**	79.1	83.7	**84.5**
Google	50.0	**72.5**	62.1	69.7	**73.4**	64.5	73.2	**75.7**	67.3	73.8	**80.3**
AOL	50.0	**71.0**	72.1	74.1	**74.3**	73.7	76.8	**77.7**	79.2	77.8	**80.7**
Reuters	50.0	**72.7**	69.7	63.3	**74.3**	75.9	63.7	**76.9**	79.5	66.7	**80.1**

Unsupervised SSFTL

Semi-supervised SSFTL

Our regression model

$$\min_{\mathbf{W}} \boxed{\Omega_{\mathcal{D}_T^\ell}(\mathbf{W})} + \lambda_1 ||\mathbf{W}||_F^2 + \lambda_2 \Omega_{\mathcal{D}_T^u}(\mathbf{W})$$

-Parameter setttings-
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 %).

Dataset	Number of source classifiers for SSFTL						
	250	500	1K	2K	5K	10K	20K
20NG	76.3	78.2	80.3	82.5	84.5	85.1	**85.6**
Google	70.6	73.1	76.6	78.5	80.3	**80.4**	80.2
AOL	67.6	76.6	78.0	78.8	80.7	**81.2**	79.1
Reuters	72.2	74.0	76.7	78.0	**80.1**	79.6	78.1

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

$$\mathbf{V}'\mathbf{F}_S^u = \mathbf{V}'\boxed{\sum_{i=1}^{k}}\varepsilon_i\mathbf{F}_{S_i}^u$$

Loss on unlabeled data

$$\Omega_{\mathbf{D}_T^u}(\mathbf{W}) = \frac{1}{n-\ell}\|\mathbf{W}'\mathbf{X}^u - \mathbf{V}'\mathbf{F}_S^u\|_F^2$$

Our regression model

$$\min_{\mathbf{W}} \Omega_{\mathcal{D}_T^\ell}(\mathbf{W}) + \lambda_1\|\mathbf{W}\|_F^2 + \lambda_2\Omega_{\mathcal{D}_T^u}(\mathbf{W})$$

-Parameter setttings-
Mode: *Semi-supervised*
Labeled target data: *20*
Unlabeled target data: *100%*
lambda_2: *0.01*

Experiments - Results

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

Dataset	Unlabeled data involved in SSFTL				
	20%	40%	60%	80%	100%
20NG	80.5	80.9	81.8	84.0	**84.5**
Google	74.5	74.9	76.4	77.9	**80.3**
AOL	73.4	75.7	77.1	78.2	**80.7**
Reuters	75.5	77.7	77.8	78.7	**80.1**

Our regression model

$$\min_{\mathbf{W}} \Omega_{\mathcal{D}_T^\ell}(\mathbf{W}) + \lambda_1 ||\mathbf{W}||_F^2 + \lambda_2 \boxed{\Omega_{\mathcal{D}_T^u}(\mathbf{W})}$$

-Parameter setttings-
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 λ_2 (accuracy in %).

Dataset	λ_2 of SSFTL						
	0	0.001	0.01	0.1	1	10	100
20NG	83.2	84.1	84.5	**85.3**	84.8	83.3	79.3
Google	76.6	79.1	**80.3**	78.7	78.2	77.4	74.3
AOL	78.3	79.5	**80.7**	79.1	78.8	76.3	73.4
Reuters	75.5	78.2	**80.1**	78.5	76.0	72.1	68.5

Supervised SSFTL

Semi-supervised SSFTL

Our regression model

$$\min_{\mathbf{W}} \Omega_{\mathcal{D}_T^\ell}(\mathbf{W}) + \lambda_1 ||\mathbf{W}||_F^2 + \boxed{\lambda_2} \Omega_{\mathcal{D}_T^u}(\mathbf{W})$$

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

Experiments - Results

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

Dataset	Uniform SSFTL				Weighted SSFTL			
	5	10	20	30	5	10	20	30
20NG	72.8	80.7	81.2	81.9	80.6	81.6	84.5	85.9
Google	64.1	67.0	70.8	77.2	73.4	75.7	80.3	81.1
AOL	69.8	71.7	72.1	74.8	74.3	77.7	80.7	82.5
Reuters	69.7	70.3	75.5	78.8	74.3	76.9	80.1	82.6

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

$$\mathbf{V}'\mathbf{F}_S^u = \mathbf{V}' \sum_{i=1}^{k} \boxed{\varepsilon_i} \mathbf{F}_{S_i}^u$$

Loss on unlabeled data

$$\Omega_{\mathbf{D}_T^u}(\mathbf{W}) = \frac{1}{n-\ell} ||\mathbf{W}'\mathbf{X}^u - \mathbf{V}'\mathbf{F}_S^u||_F^2$$

Our regression model

$$\min_{\mathbf{W}} \Omega_{\mathcal{D}_T^\ell}(\mathbf{W}) + \lambda_1 ||\mathbf{W}||_F^2 + \lambda_2 \Omega_{\mathcal{D}_T^u}(\mathbf{W})$$

-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.

Transfer learning methods	*Scalability*	*Diff. label*
RSP [Shi *et al.*, 2009]	×	√
EigenTransfer [Dai *et al.*, 2009]	×	√
MTL-MI [Quadrianto *et al.*, 2010]	×	√
DAM [Duan *et al.*, 2009]	√	×
LWE [Gao *et al.*, 2008]	√	×
SSFTL	√	√

Summary

- ***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.*

Heterogeneous Transfer Learning

Heterogeneous Transfer Learning for Image Clustering via the Social Web.

Qiang Yang, Yuqiang Chen, Gui-Rong Xue, Wenyuan Dai and Yong Yu.

In *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP* (**ACL-IJCNLP'09**), Singapore, Aug 2009, pages 1 -- 9.

HTL Setting: Text to Images

- Source data: labeled or unlabeled
- Target training data: labeled

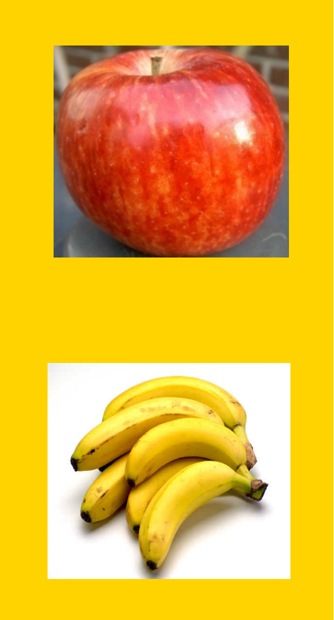
Apple

The apple is the pomaceous fruit of the apple tree, species Malus domestica in the rose family Rosaceae ...

Banana

Banana is the common name for a type of fruit and also the herbaceous plants of the genus Musa which produce this commonly eaten fruit ...

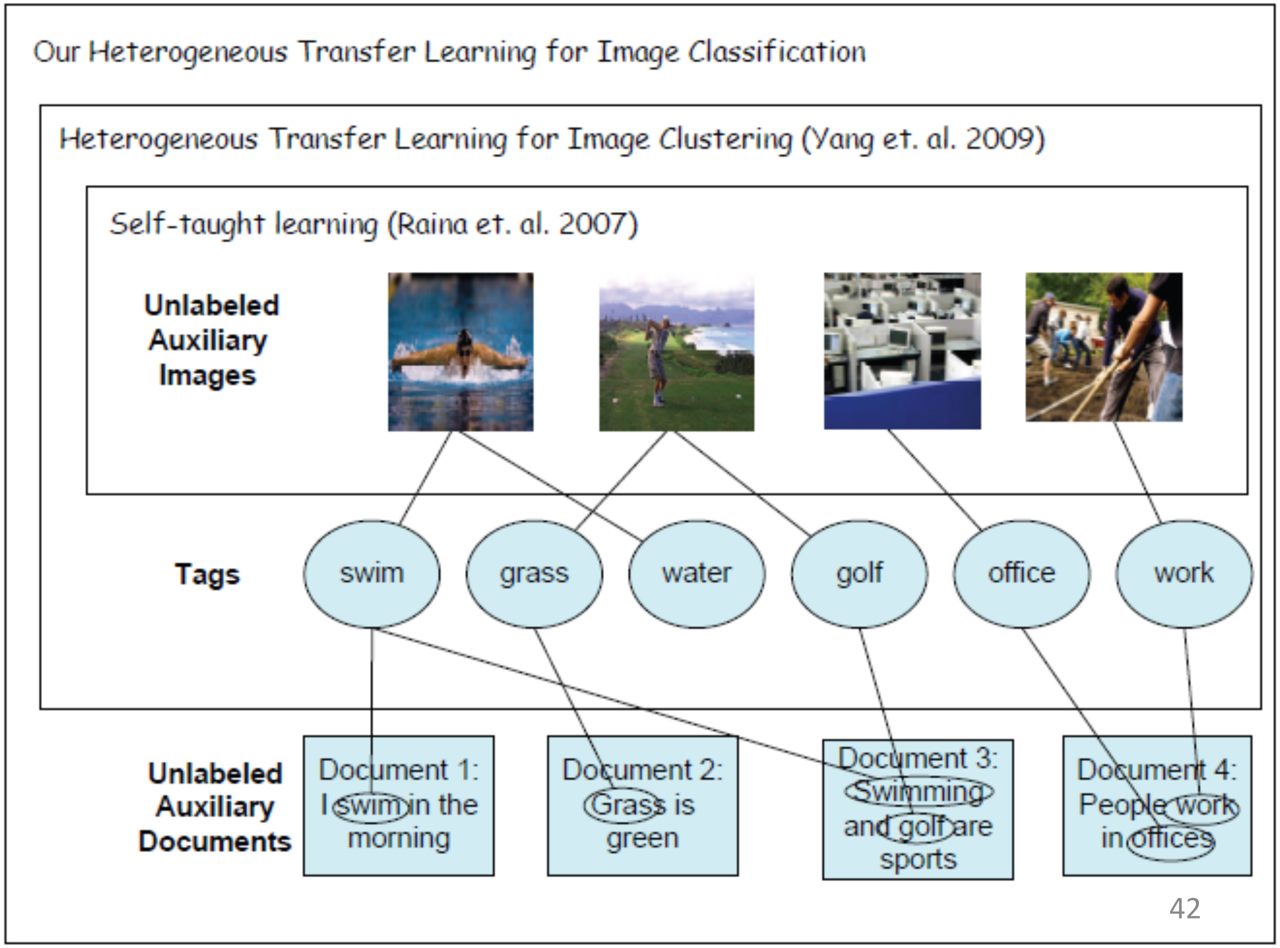
Training: Text



Testing: Images

Y. Zhu, G. Xue, Q. Yang et al. Heterogeneous transfer learning for image classification. AAAI 2011

Unlabeled Source data



Current Work on HTL - Clustering

- Core idea:
  - Looking for a latent space Z (cluster center space)

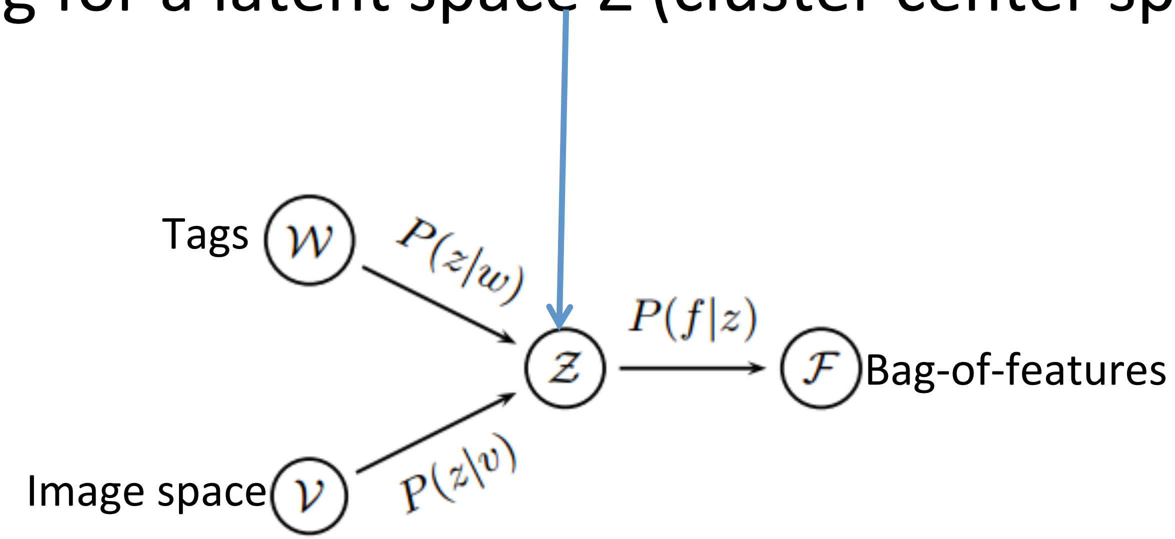
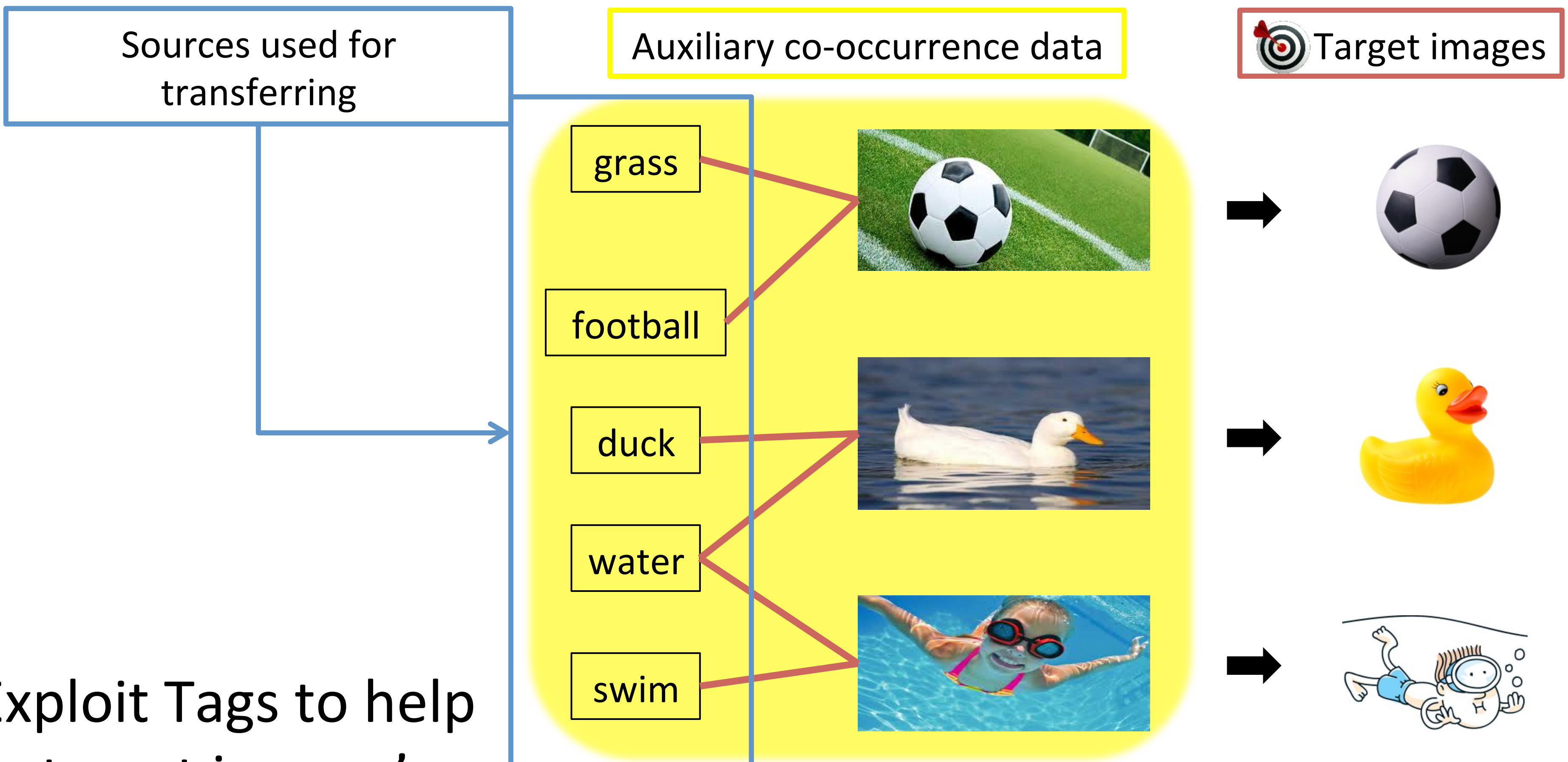


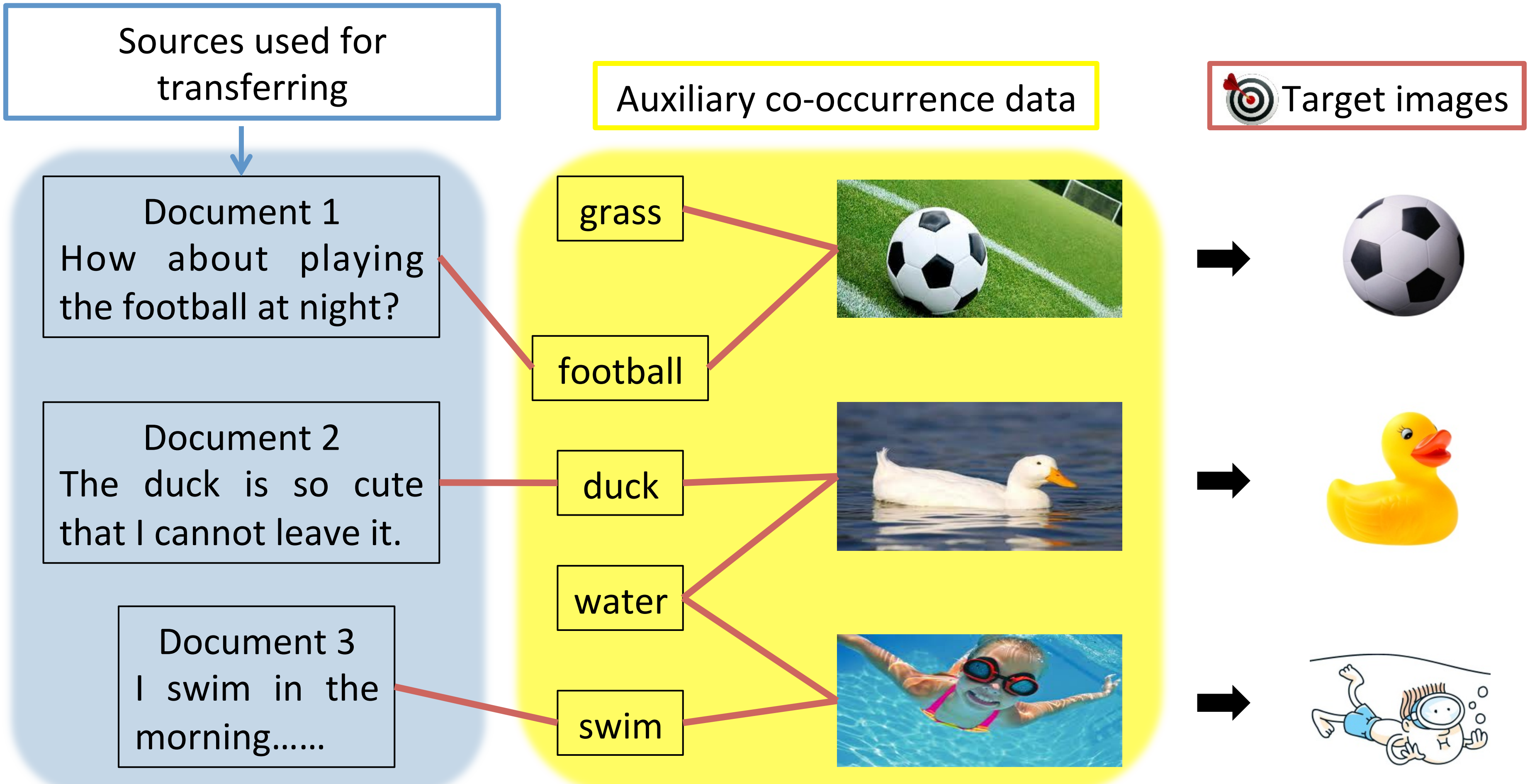
Figure 3: Graphical model representation of aPLSA model.

Current Work on HTL - Clustering



Exploit Tags to help target images' clustering

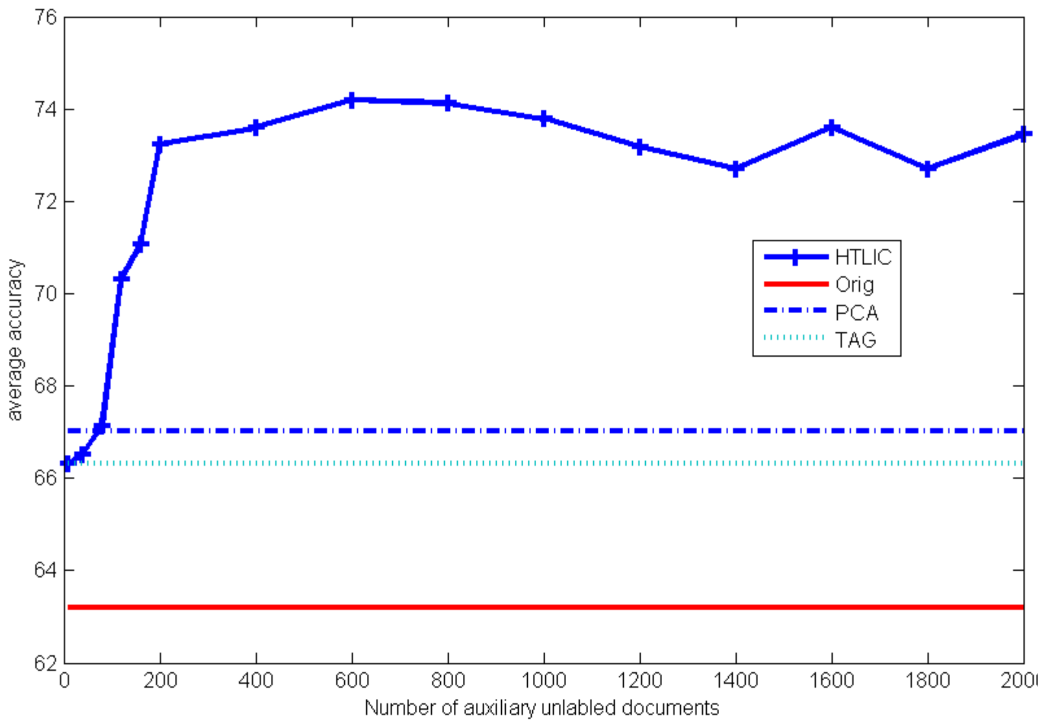
Current Work on HTL - Classification



Exploit abundant unlabeled documents to help target images' classification

Experiments: # text docs

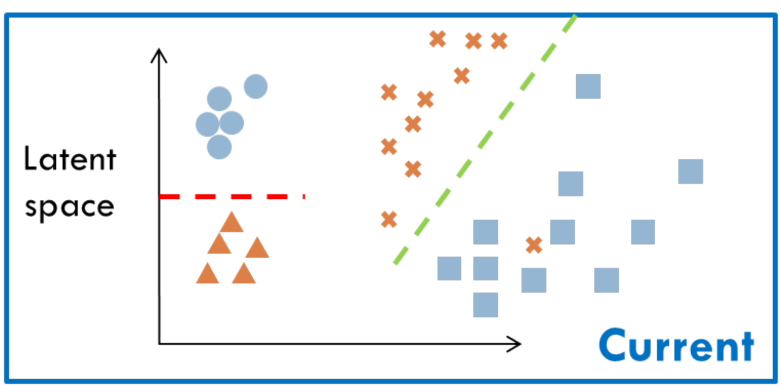
Accuracy



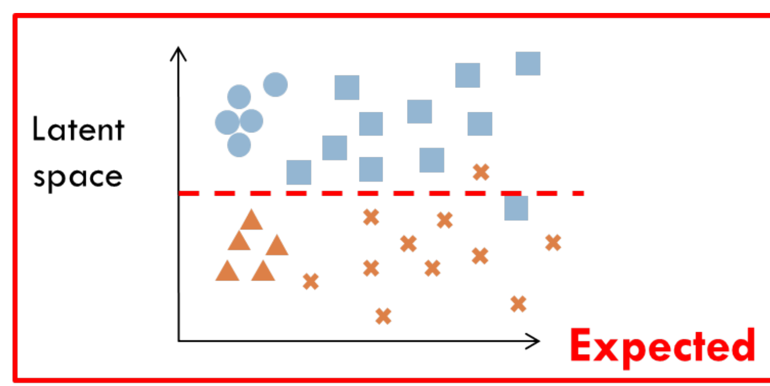
text docs

Adding documents as if they were images (Ying Wei and Yangqiu Song)

- Supervised Alignment and Classification
  - Obtain the latent space as Yin's work, i.e. CMF
  - Project both source and target data into the latent space, as depicted in figure (a)
  - Align and classify simultaneously, obtain the results in figure (b)



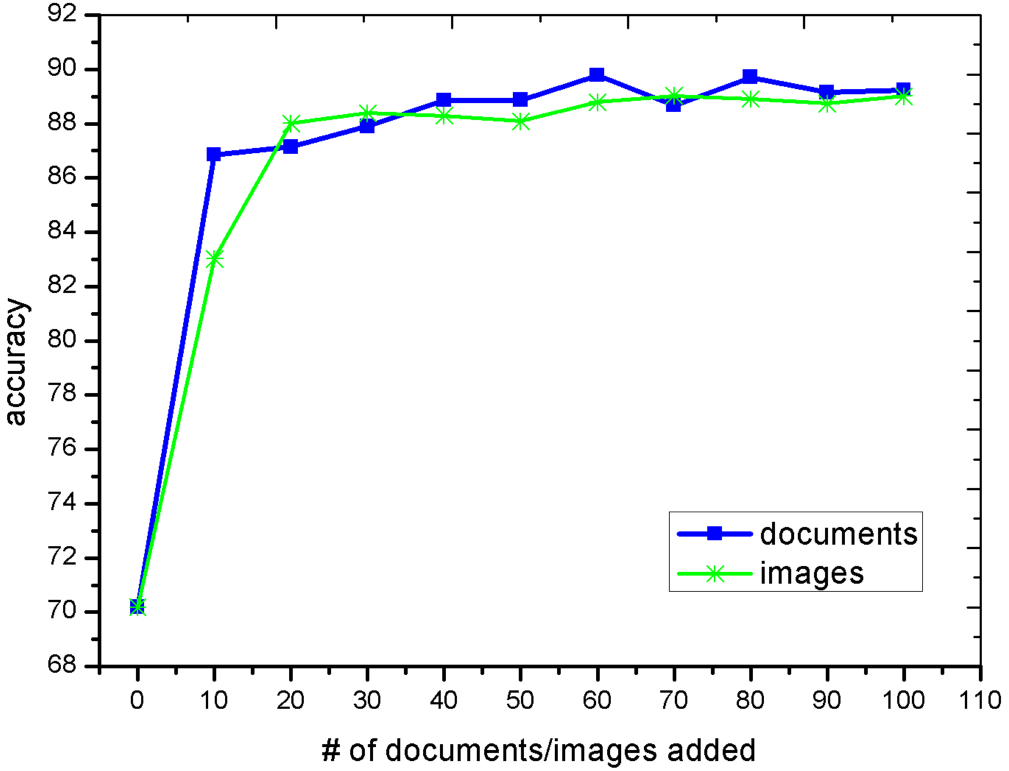
(a)



(b)

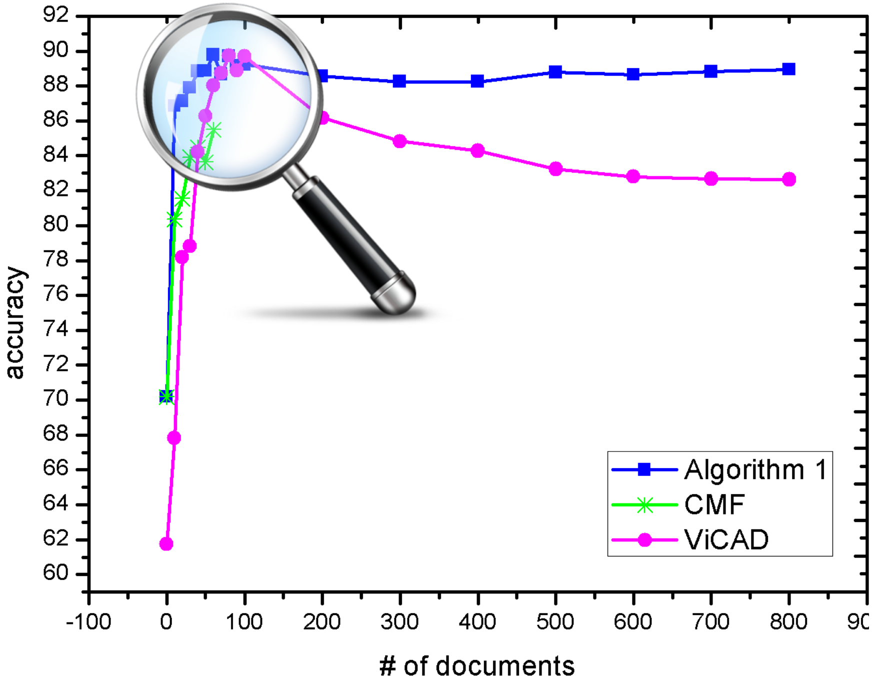
Results

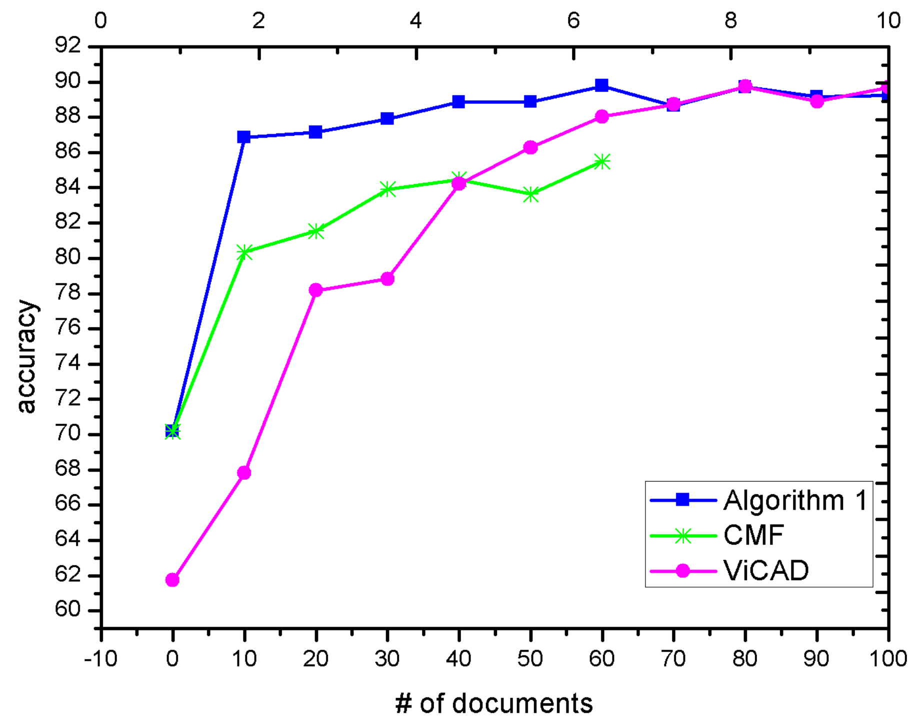
- why add documents/ not images?
  - Abundant documents but comparably less labeled images
  - The documents added may outperform the same number of images added



Results

- Comparison of Algorithm 1/CMF/ViCAD
  - CMF can hardly converge after 60 documents added





Conclusions

- We have seen three applications of Transfer Learning
  - cross-domain sensor-based activity recognition
  - social-media source free transfer learning
  - Heterogeneous transfer learning