迁移学习理论与应用
Transfer Learning: An Overview

杨强，香港科大
Qiang Yang, HKUST

Thanks:
Sinno Jialin Pan, NTU, Singapore
Ying Wei, HKUST, Hong Kong
Ben Tan, HKUST, Hong Kong

A psychological point of view

- **Transfer of Learning** (学习迁移）in Education and Psychology
  - 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.
- E.g.
  - C++ → Java
  - Math/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 transfer knowledge to new domains/tasks (target)?

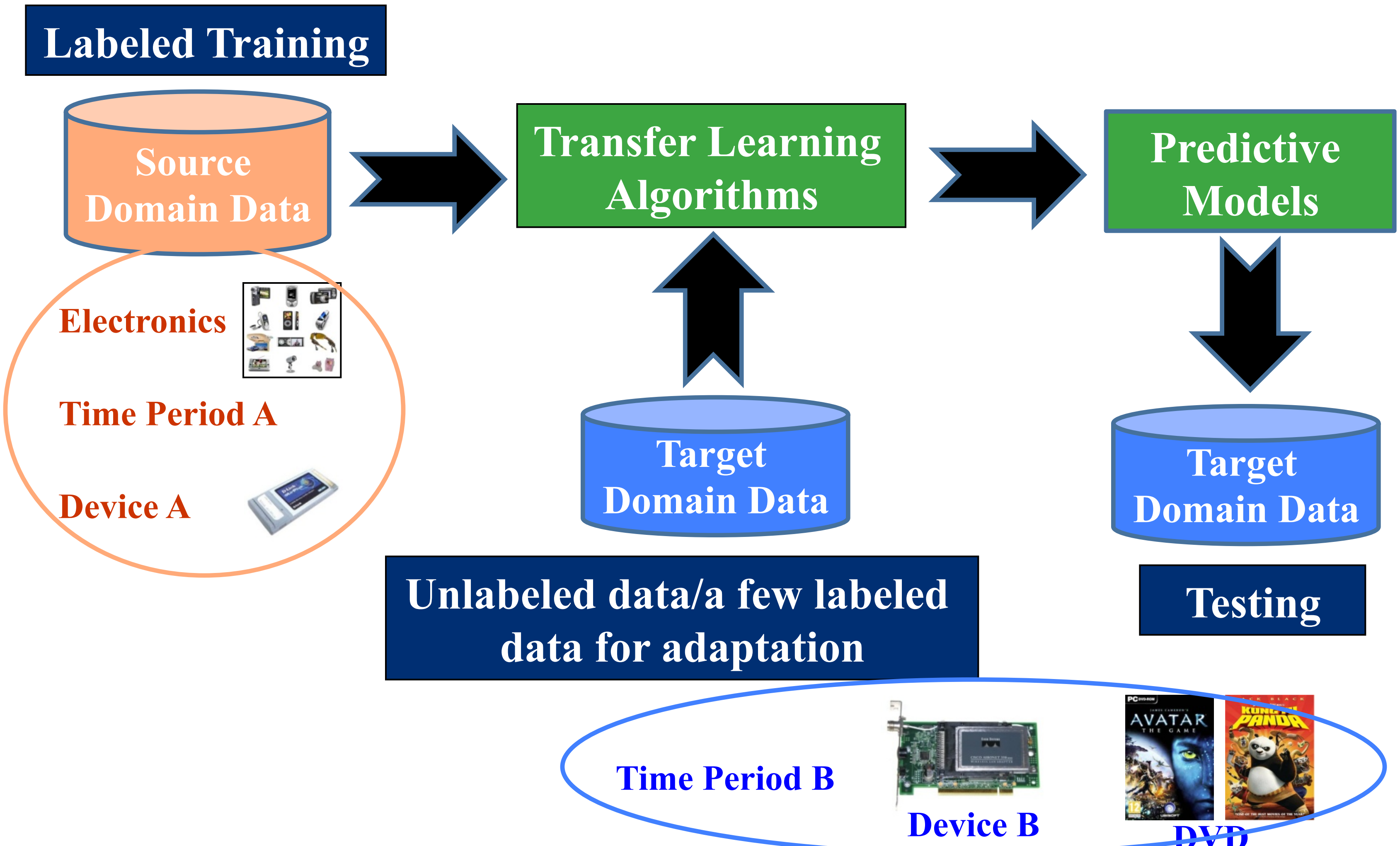
- Key:
  – Representation Learning, Change of Representation

Why Transfer?

- Build every model from scratch?
  - Time consuming and expensive
  - Expense:
    - Data Collection/Labeling
    - Privacy
    - Time to train

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

Why Transfer Learning?



Transfer Learning

Different fields

- Transfer learning for reinforcement learning.

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

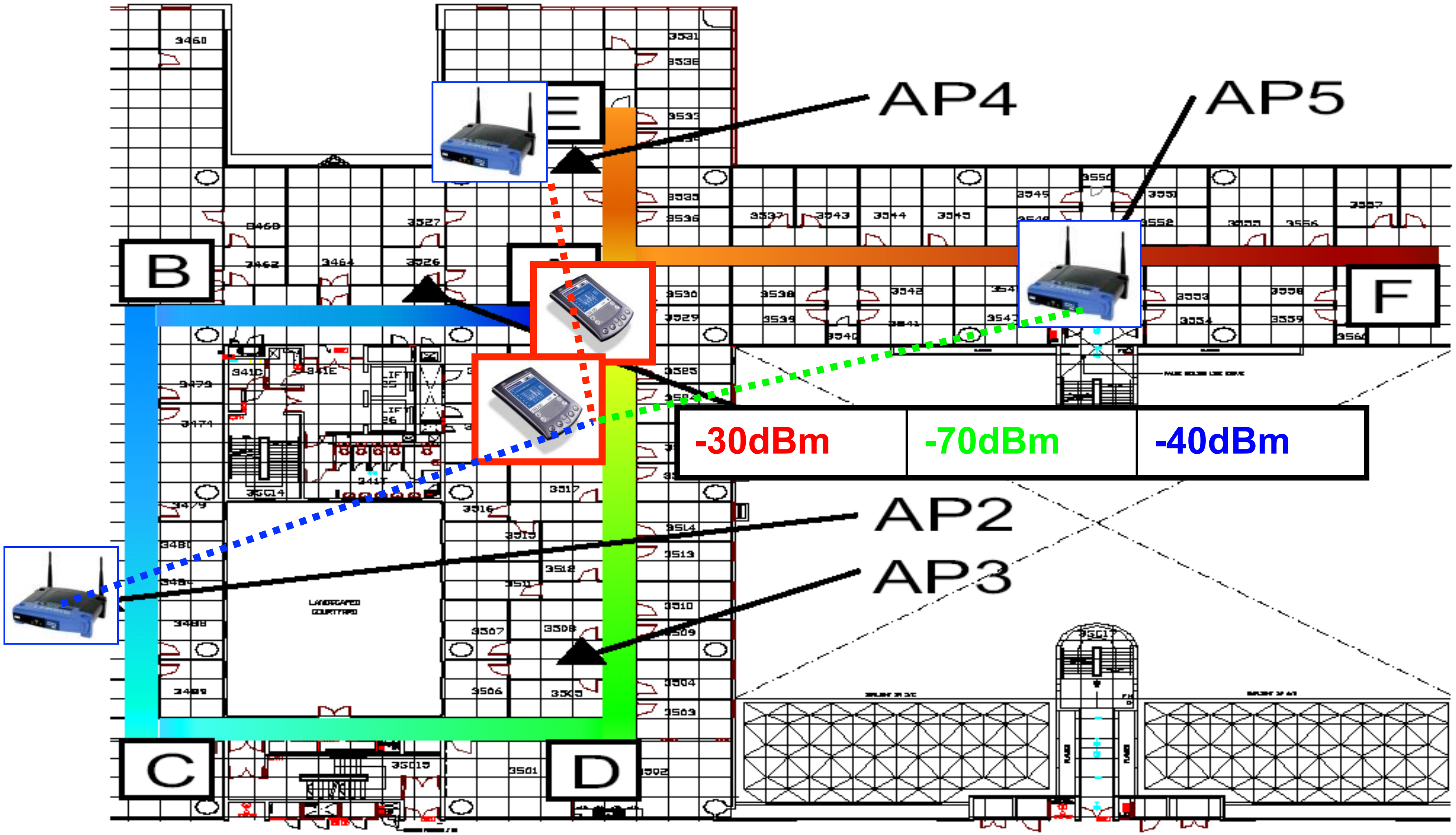
- Transfer learning for classification, and regression problems.

Focus!

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

Motivating Example I:

Indoor WiFi localization



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)

Device A

Localization model

Test

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

Device A

~ 1.5 meters

Drop!

Training

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

Device B

Localization model

Test

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

Device A

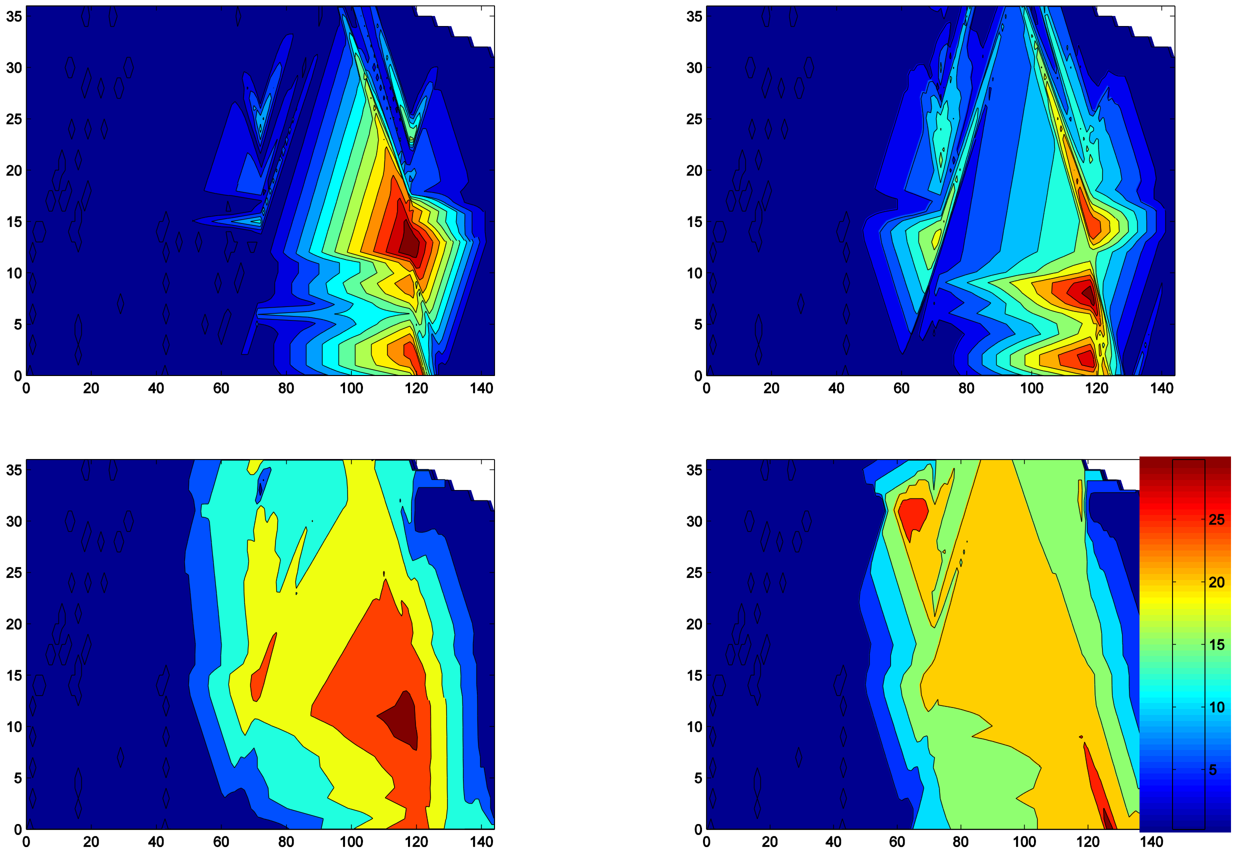
~10 meters

Difference between Domains

Time Period A | Time Period B

Device A

Device B



Motivating Example II:

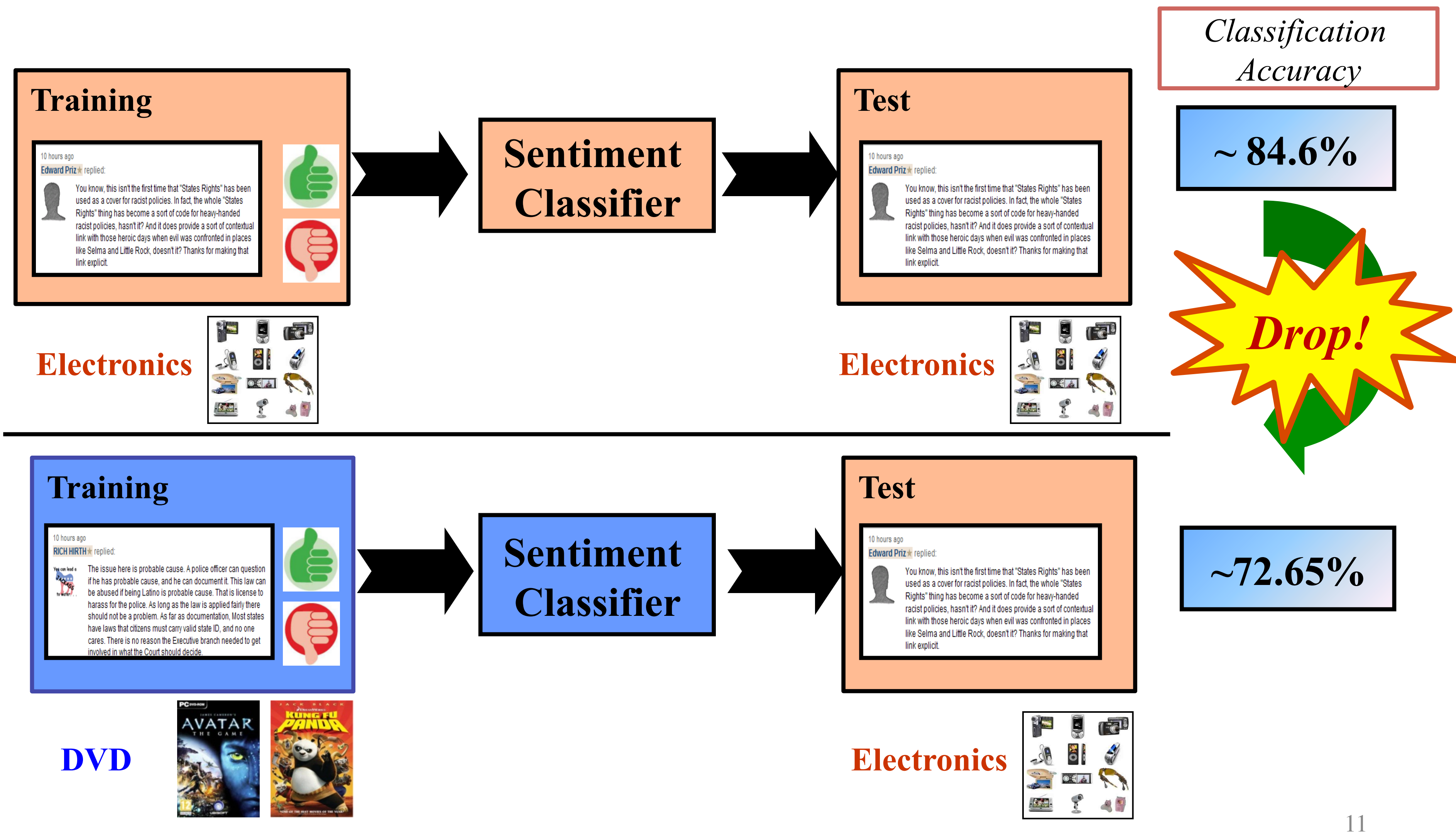
Sentiment classification

10 hours ago
Edward Priz replied:
You know, this isn't the first time that "States Rights" has been used as a cover for racist policies. In fact, the whole "States Rights" thing has become a sort of code for heavy-handed racist policies, hasn't it? And it does provide a sort of contextual

10 hours ago
RICH HIRTH replied:
The issue here is probable cause. A police officer can question if he has probable cause, and he can document it. This law can be abused if being Latino is probable cause. That is license to harass for the police. As long as the law is applied fairly there

2 hours ago
Julia Gomez replied:
The Arizona law is so clearly unconstitutional that I do not think it will ever reach the point of being enforced. The article did not say so, but the Republican governor is afraid of a GOP primary electorate that is even more reactionary than usual. That is why she signed the bill, not because she thinks it is legally defensible.

Sentiment Classification (cont.)



Difference in Representation



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.

A Major Assumption in Traditional Machine Learning

- Training and future (test) data come from the same domain, which implies
  - Represented in the same feature spaces.
  - Follow the same data distribution.

Machine Learning: Yesterday, Today and Tomorrow

Yesterday Today Tomorrow

Machine Learning: Yesterday, Today and Tomorrow

Yesterday Today Tomorrow

Different Scenarios

- Training and testing data may come from different domains:
  - ❑Different different feature spaces/ marginal distributions:

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

  - ❑Different conditional distributions or different label spaces:

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

Transfer Learning Approaches

Instance-based Approaches

Feature-based Approaches

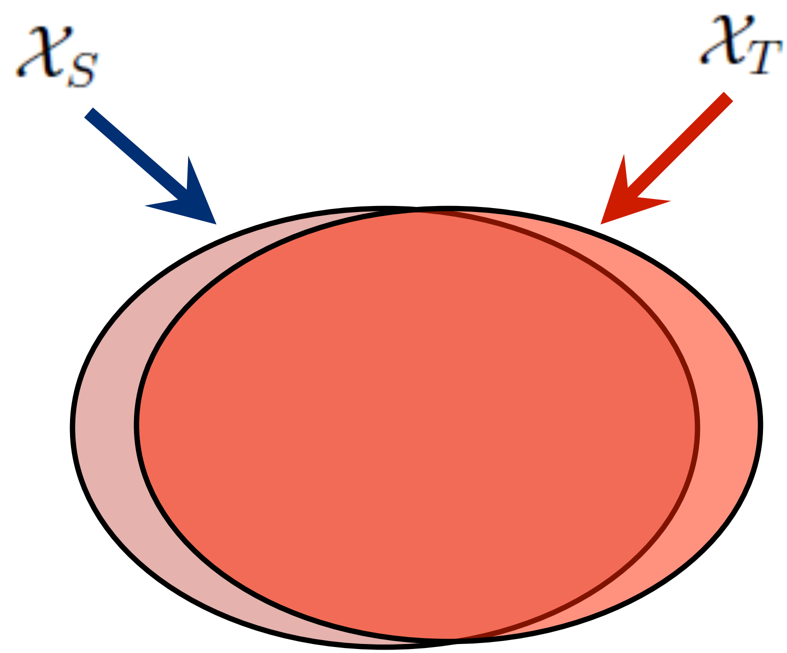
Parameter/Model - based Approaches

Relational Approaches

Instance-based Transfer Learning Approaches

General Assumption

Source and target domains have a lot of overlapping features



Instance-based Transfer Learning Approaches

Case I: Unlabeled Target

Problem Setting

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

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: Some Labels in Target

Problem Setting

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

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

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,

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

Instance-based Approaches

Case I (cont.)

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

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

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

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

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 (cont.)

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

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

$$\Downarrow$$

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

[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

Direct density ratio estimation

[Sugiyama *etal*., NIPS-07, Kanamori *etal*., JMLR-09]

Recall $\beta(x) = \dfrac{P_T(x)}{P_S(x)}$

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

KL divergence loss

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

[Sugiyama *etal*., NIPS-07]

Least squared loss

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

[Kanamori *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

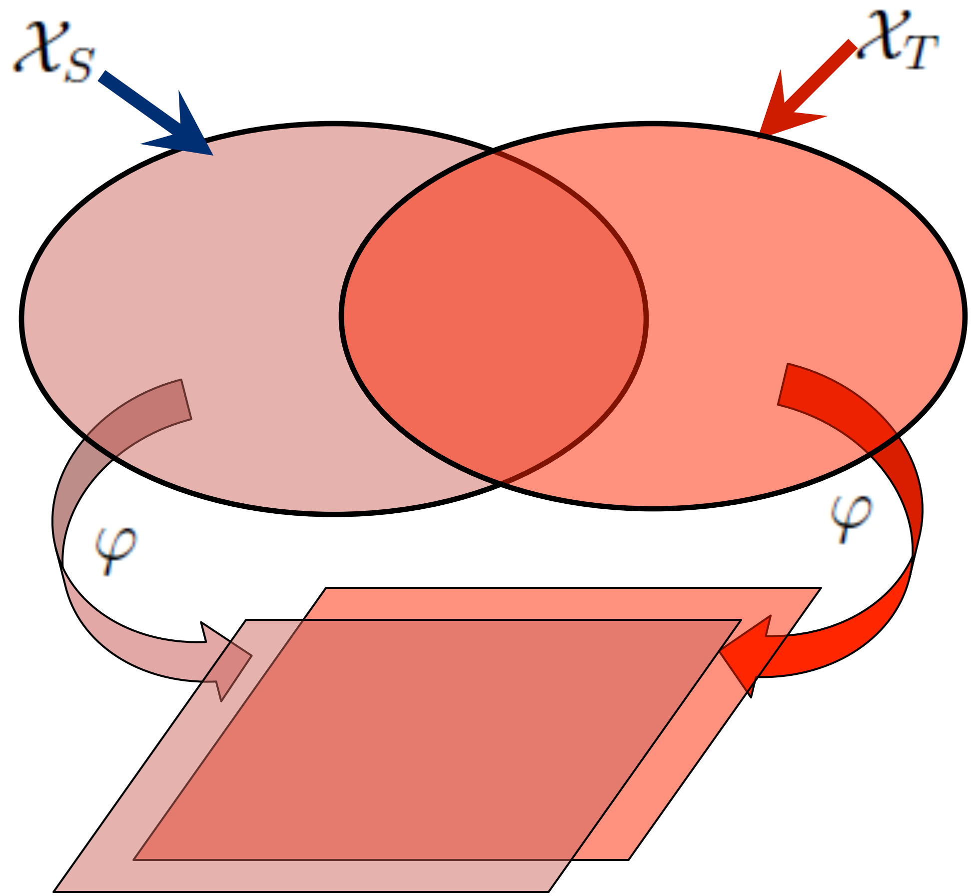
Instance-based Approaches

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.

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

	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.

Feature-based Approaches

Encode application-specific knowledge (cont.)

Electronics

	compact	sharp	blurry	hooked	realistic	boring
	1	1	0	0	0	0
	0	1	0	0	0	0
	0	0	1	0	0	0

Training

$$y = f(x) = \mathrm{sgn}(w \cdot x^T), \qquad w = [1,1,-1,0,0,0]$$

Prediction

Video Game

	compact	sharp	blurry	hooked	realistic	boring
	0	0	0	1	0	0
	0	0	0	1	1	0
	0	0	0	0	0	1

Feature-based Approaches

Encode application-specific knowledge (cont.)

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.

Feature-based Approaches

Encode application-specific knowledge (cont.)

- Three different types of features
  - Source domain (***Electronics***) specific features, e.g., ***compact, sharp, blurry***
  - Target domain (***Video Game***) specific features, e.g., ***hooked, realistic, boring***
  - Domain independent features (pivot features), e.g., ***good, excited, nice, never_buy***

Feature-based Approaches

Encode application-specific knowledge (cont.)

- How to identify ***pivot*** features?
  - Term frequency on both domains
  - Mutual information between features and labels (source domain)
  - Mutual information on between features and domains

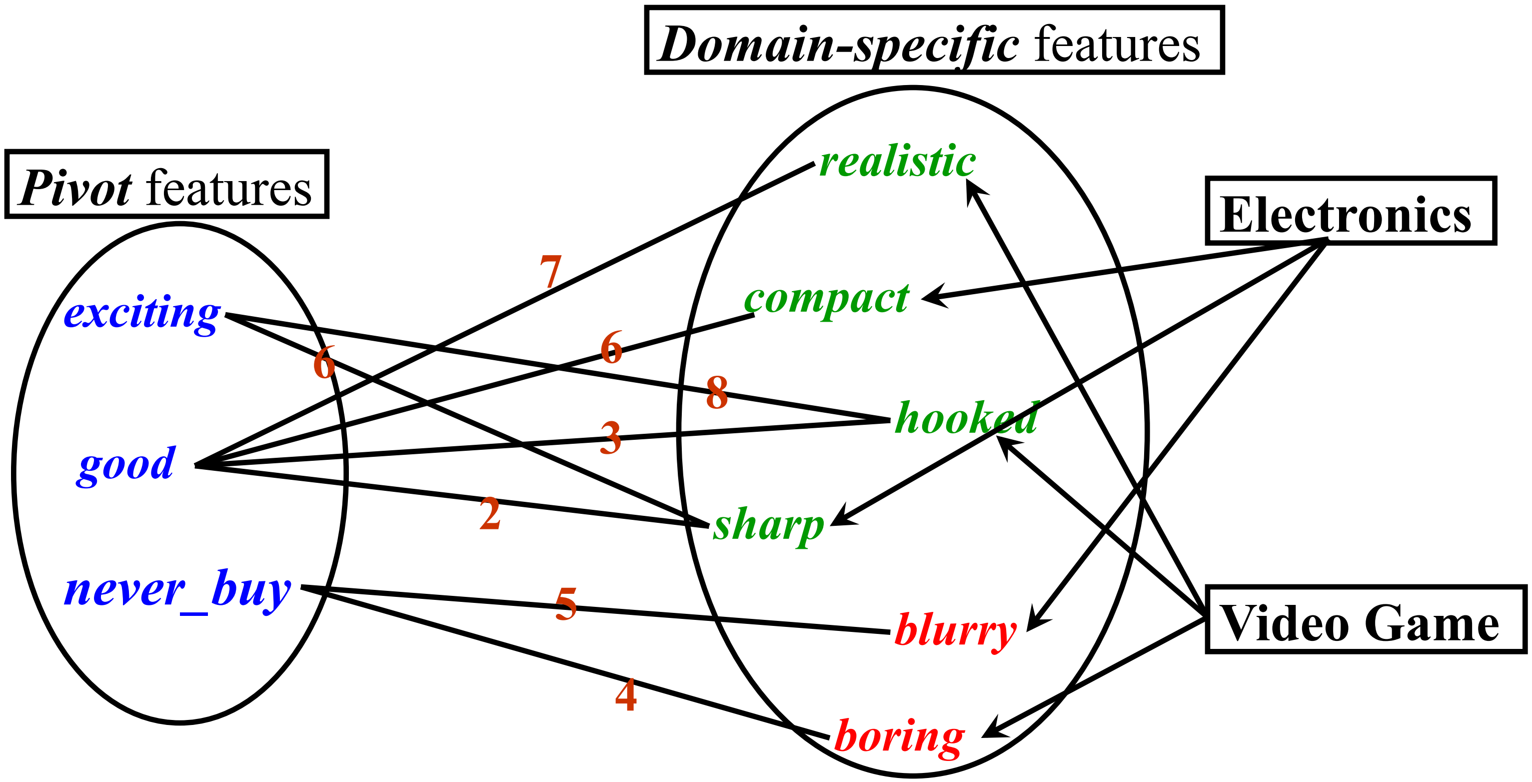
- How to utilize pivots to ***align*** features across domains?
  - Structural Correspondence Learning (SCL) [Biltzer *etal*. EMNLP-06]
  - Spectral Feature Alignment (SFA) [Pan *etal*. WWW-10]

Feature-based Approaches Spectral Feature Alignment (SFA)

- **Intuition**
  - Use a ***bipartite*** graph to model the correlations between ***pivot*** features and other features
  - Discover new shared features by applying ***spectral clustering*** techniques on the graph

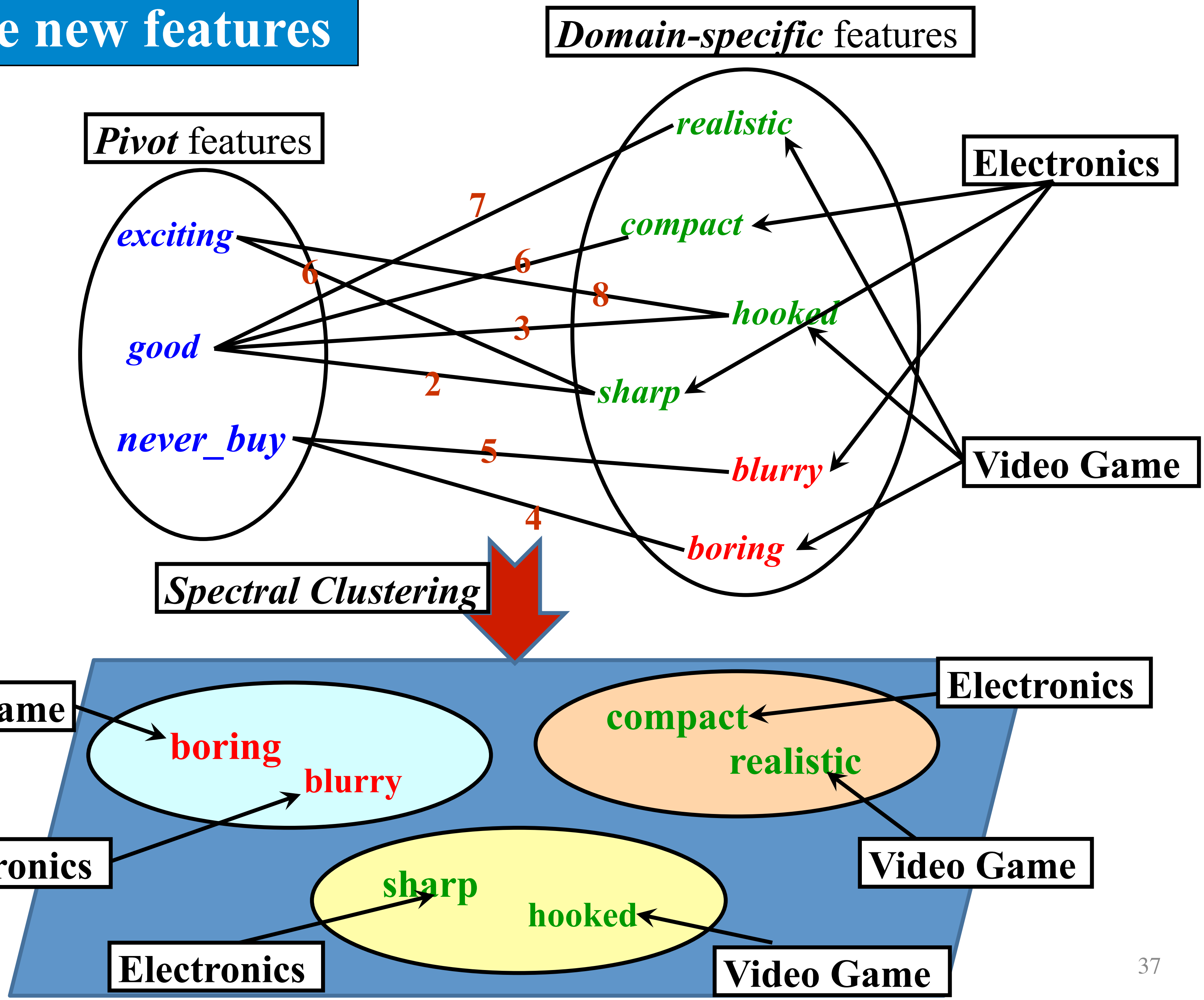
Spectral Feature Alignment (SFA)

High level idea



- If two ***domain-specific*** words have connections to more common ***pivot*** words in the graph, they tend to be aligned or clustered together with a higher probability.
- If two ***pivot*** words have connections to more common ***domain-specific*** words in the graph, they tend to be aligned together with a higher probability.

Derive new features



Spectral Feature Alignment (SFA)

Derive new features (cont.)

Electronics

	sharp/hooked	compact/realistic	blurry/boring
	1	1	0
	1	0	0
	0	0	1

Training

$$y = f(x) = \mathrm{sgn}(w \cdot x^T), \qquad w = [1, 1, -1]$$

Prediction

Video Game

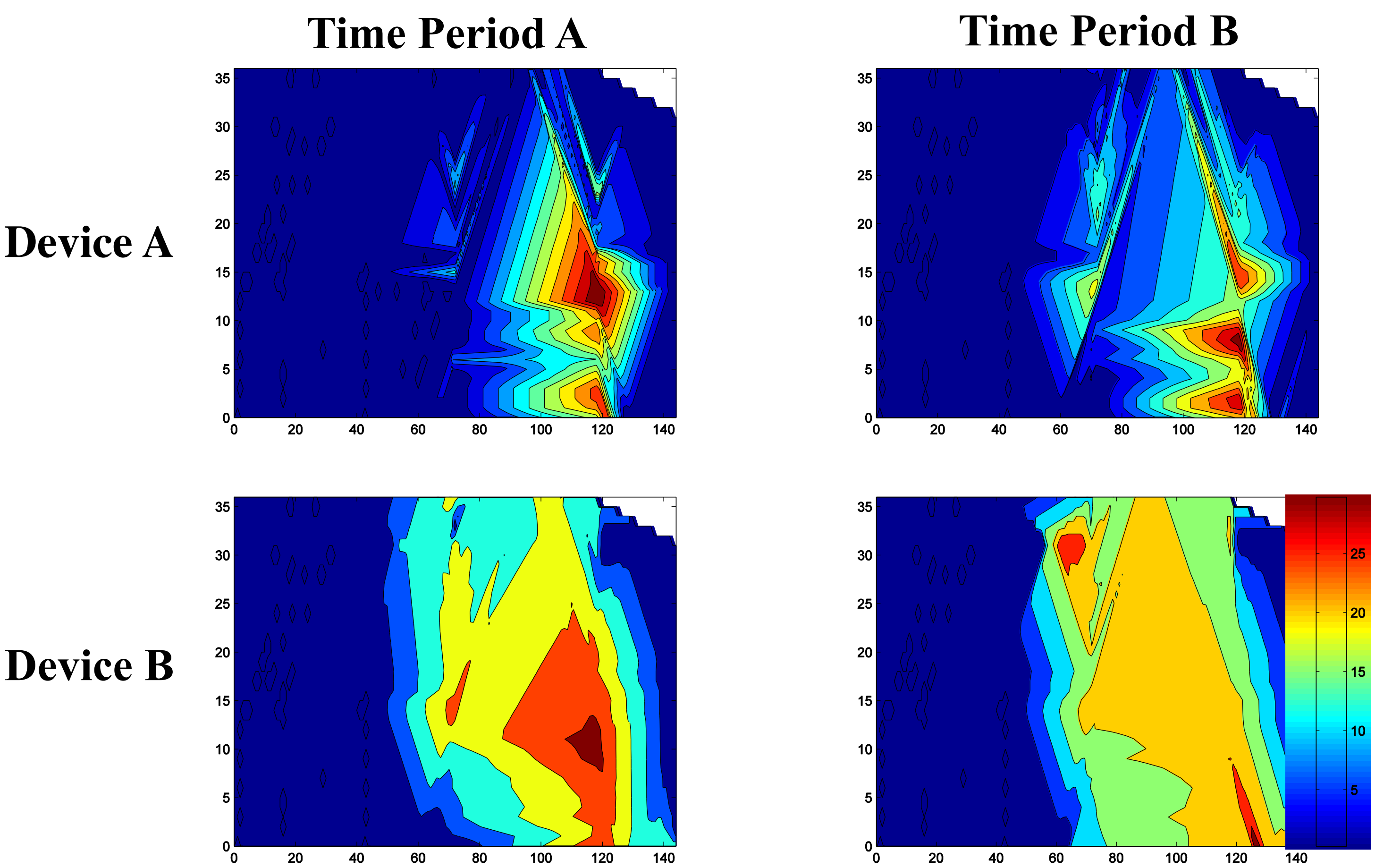
	sharp/hooked	compact/realistic	blurry/boring
	1	0	0
	1	1	0
	0	0	1

Spectral Feature Alignment (SFA)

1. Identify P ***pivot*** features
2. Construct a ***bipartite*** graph between the pivot and remaining features.
3. Apply ***spectral clustering*** on the graph to derive new features
4. Train classifiers on the source using ***augmented*** features (original features + new features)

Feature-based Approaches

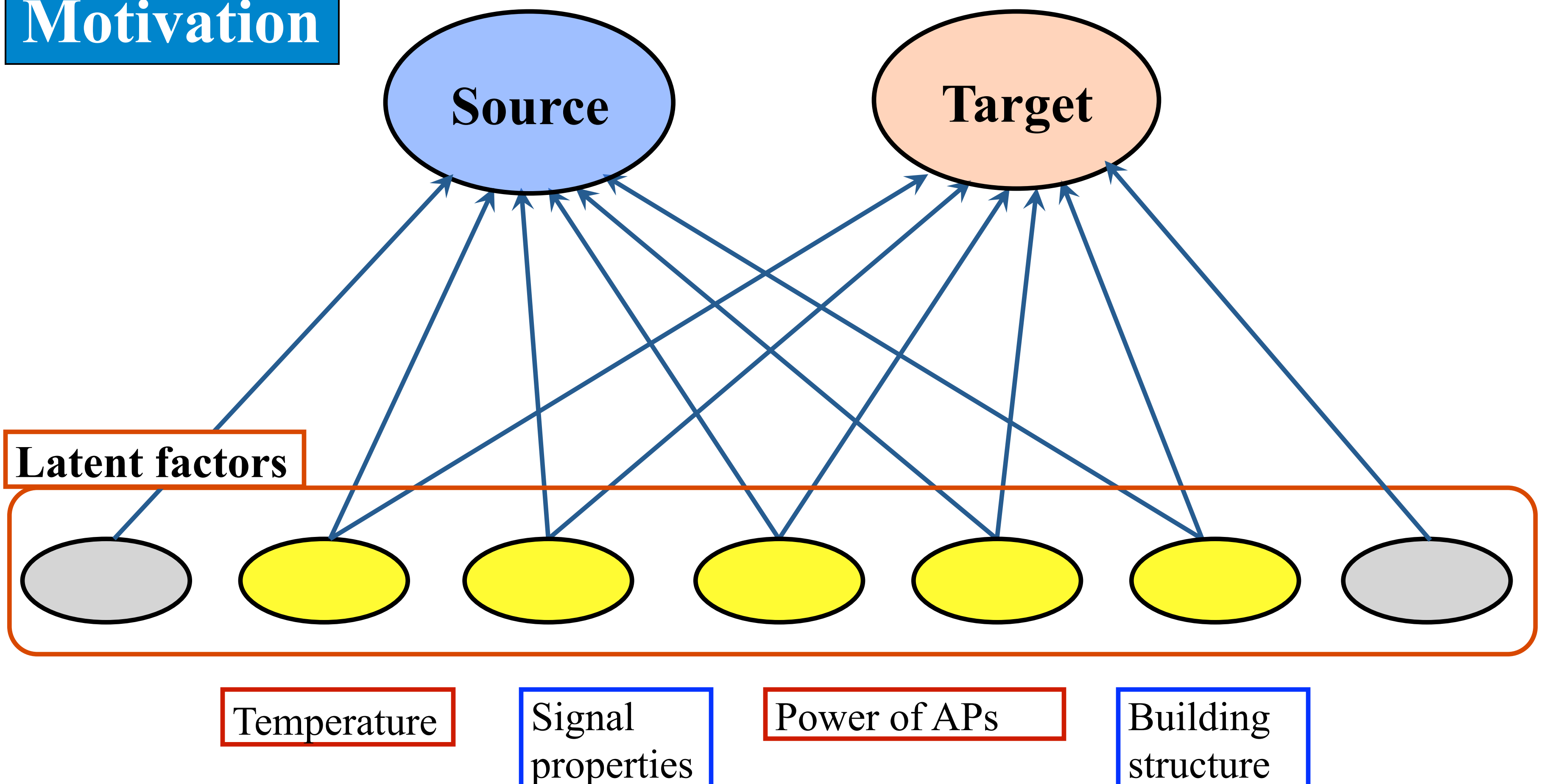
Develop general approaches



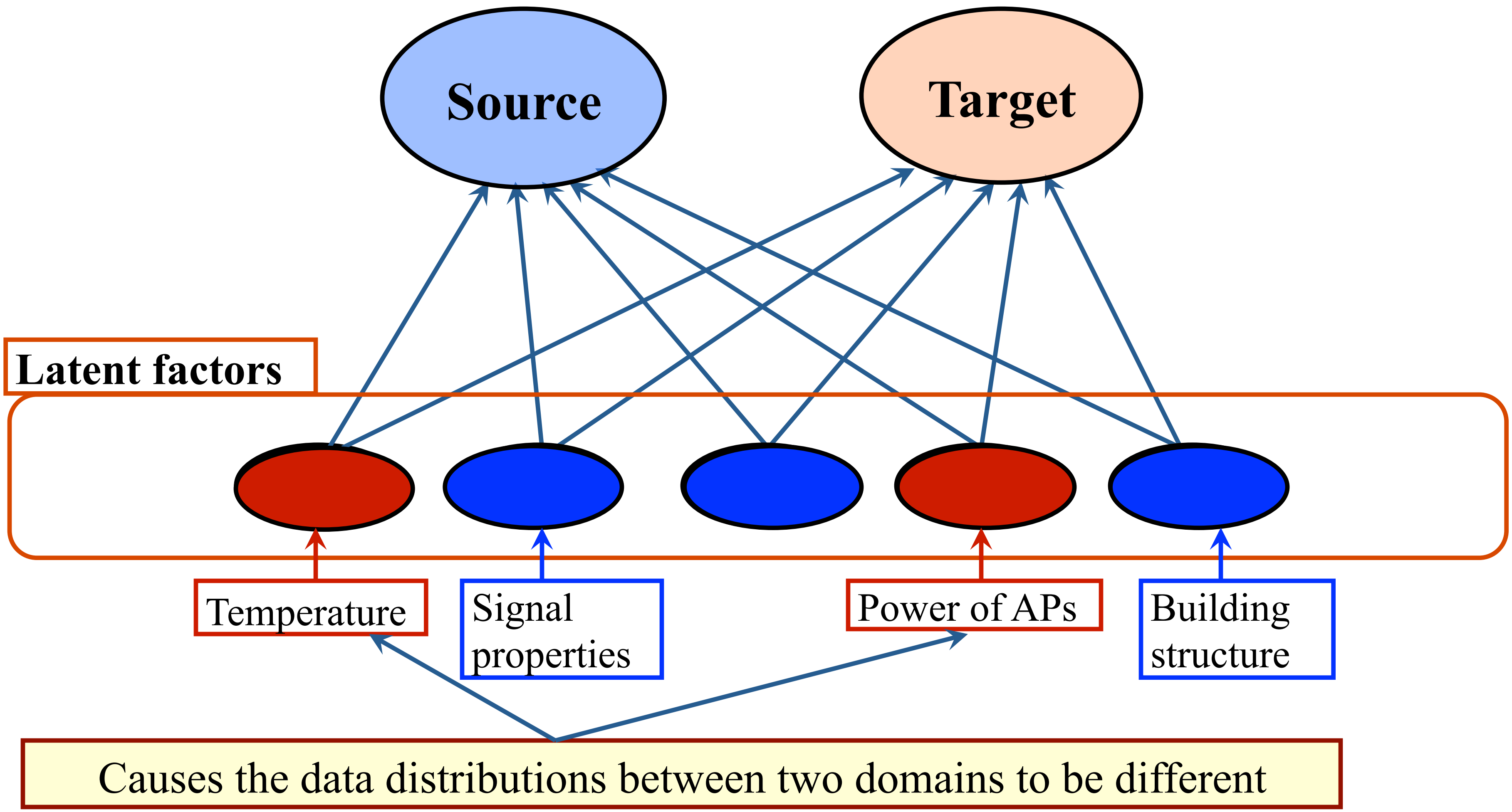
Feature-based Approaches

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

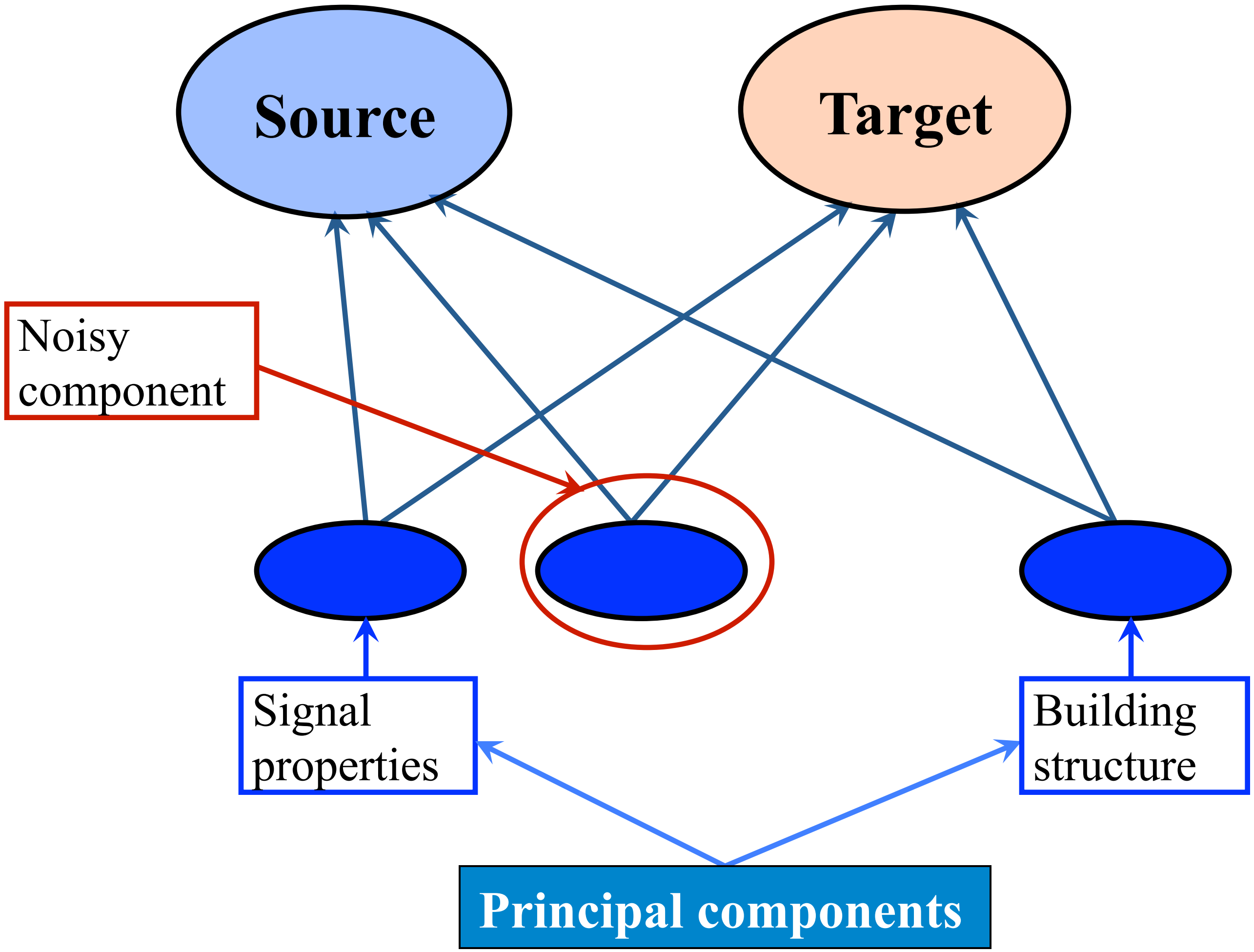
Motivation



Transfer Component Analysis (cont.)

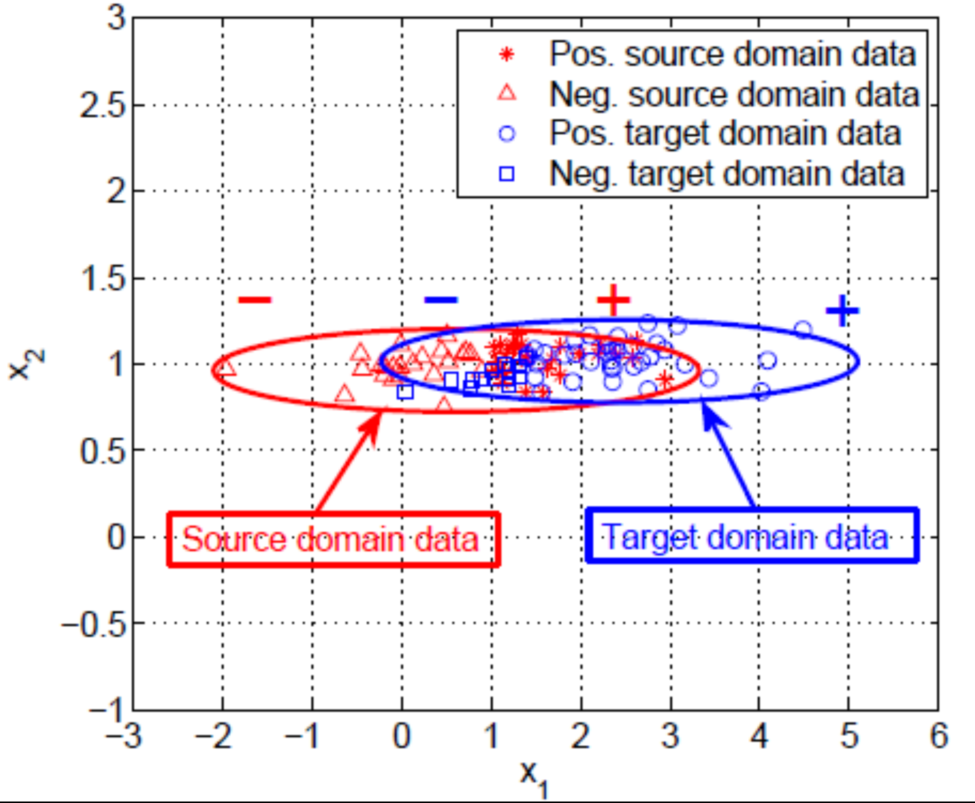


Transfer Component Analysis (cont.)



Transfer Component Analysis (cont.)

Learning φ by only minimizing the distance between distributions



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 difference and preserve original data structure.

High level optimization problem

$$\min_{\varphi} \quad \mathrm{Dist}(\varphi(\mathbf{X}_S), \varphi(\mathbf{X}_T)) + \lambda\Omega(\varphi)$$

$$\text{s.t.} \quad \text{constraints on } \varphi(\mathbf{X}_S) \text{ and } \varphi(\mathbf{X}_T)$$

Transfer Component Analysis (cont.)

Recall: Maximum Mean Discrepancy (MMD)

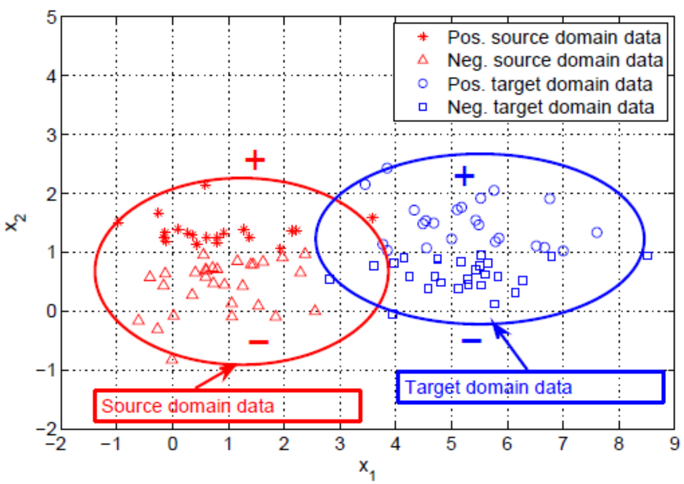
Given $\mathbf{X}_S = \{x_{S_i}\}_{i=1}^{n_S}$, $\mathbf{X}_T = \{x_{T_i}\}_{i=1}^{n_T}$, drown from $P_S(x)$ and $P_T(x)$, respectively,

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

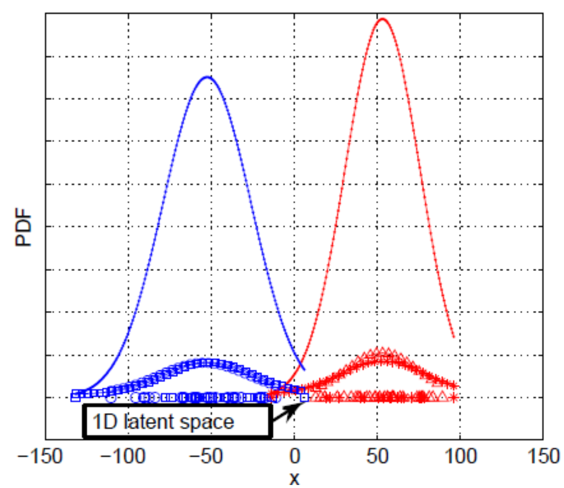
Transfer Component Analysis (cont.)

An illustrative example

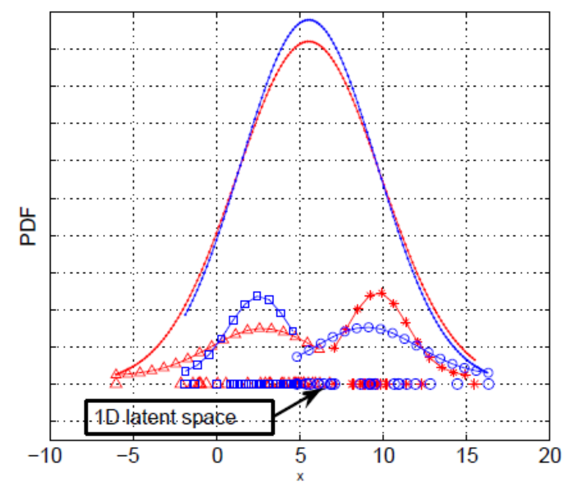
Latent features learned by **PCA** *and* **TCA**



Original feature space



PCA



TCA

Feature-based Approaches

Self-taught Feature Learning (Andrew Ng. et al.)

- **Intuition:** Useful higher-level features can be learned from unlabeled data.
- **Steps:**

1) Learn higher-level features from a lot of unlabeled data.
2) Use the learned higher-level features to represent the data of the target task.
3) Train models from the new representations of the target task (supervised)

- **How to learn higher-level features**
  - Sparse Coding [Raina etal., 2007]
  - Deep learning [Glorot *etal.*, 2011]

Feature-based Approaches

Multi-task Feature Learning

General Multi-task Learning Setting

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

where n_S and n_T are small,

Learn f_S, f_T, s.t. $\sum_{t \in \{S,T\}} \sum_i \epsilon(f_t(x_{t_i}), y_{t_i})$ is small.

- **Assumption:** If tasks are related, they should share some **good** common features.
- **Goal:** Learn a low-dimensional representation shared across related tasks.

Multi-task Learning

Assumption:

If tasks are related, they may share similar parameter vectors.

For example, [Evgeniou and Pontil, KDD-04]

Common part

$$\theta_S = \theta_0 + v_S$$

$$\theta_T = \theta_0 + v_T$$

Specific part for individual task

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

Multi-task Feature Learning

Assume $f(x) = \langle \theta, (U^\top x) \rangle = \theta^\top (U^\top x)$, where $\theta \in \mathbb{R}^k, x \in \mathbb{R}^m, U \in \mathbb{R}^{m \times k}$

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

$$\text{s.t.} \quad \text{constraints on } U.$$

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

$$\begin{cases} U \text{ is full rank } (U \in \mathbb{R}^{m \times k}, k = m), \Theta \text{ is sparse.} \\ U \text{ is low rank } (U \in \mathbb{R}^{m \times k}, k \ll m). \end{cases}$$

[Argyriou *etal.*, NIPS-07]

[Ando and Zhang, JMLR-05]

[Ji *etal*, KDD-08]

Deep Learning in Transfer Learning

Transfer Learning with Deep Learning

Transfer Learning Perspective:
Why need Deep Learning?

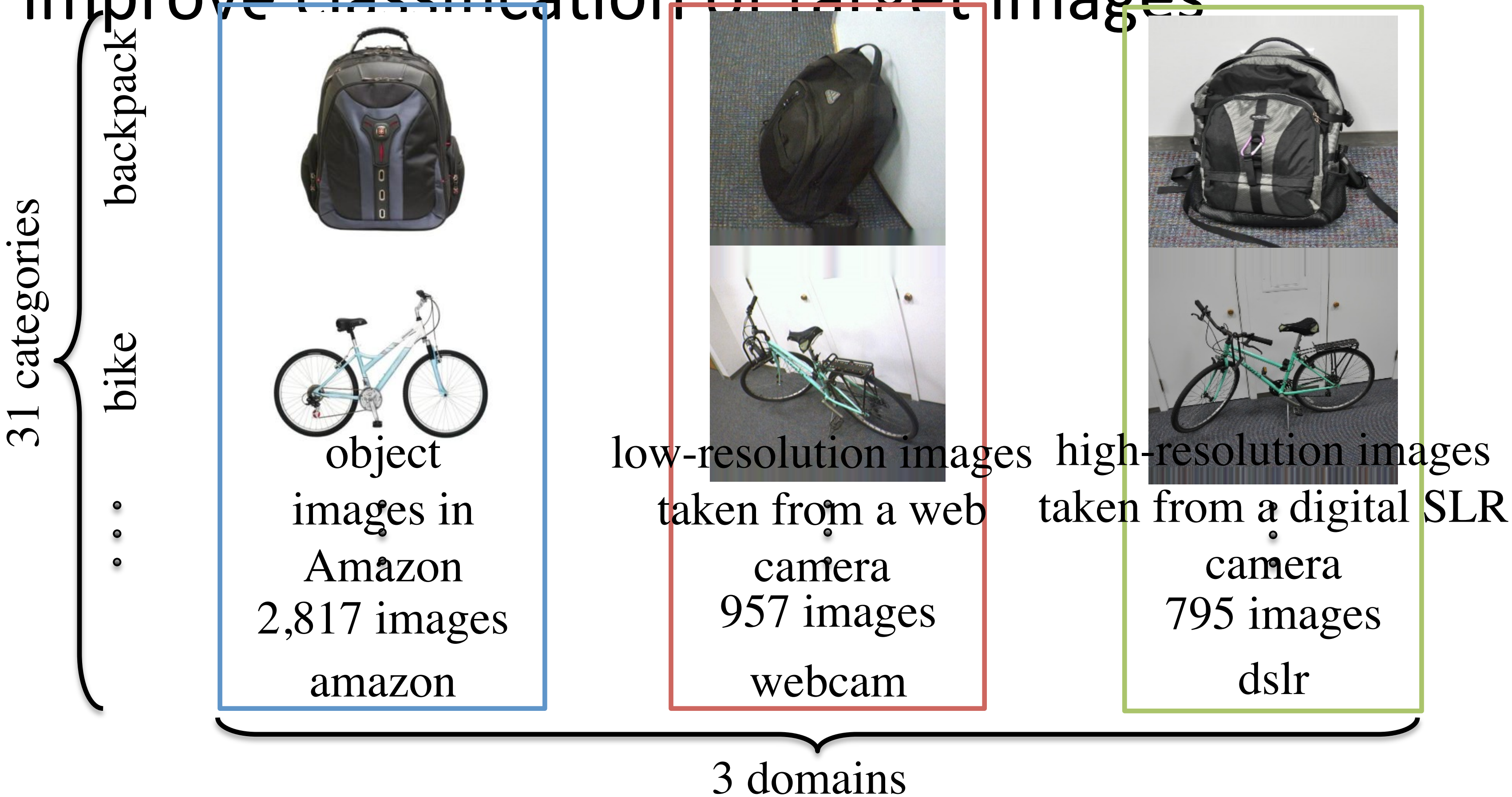
- Deep neural networks learn nonlinear representations
  - that are hierarchical;
  - that disentangle different explanatory factors of variation behind data samples;
  - that manifest invariant factors underlying different populations.

Deep Learning Perspective:
Why need Transfer Learning?

- Transfer Learning alleviates
  - the incapability of learning on a dataset which may not be large enough to train an entire deep neural network from scratch

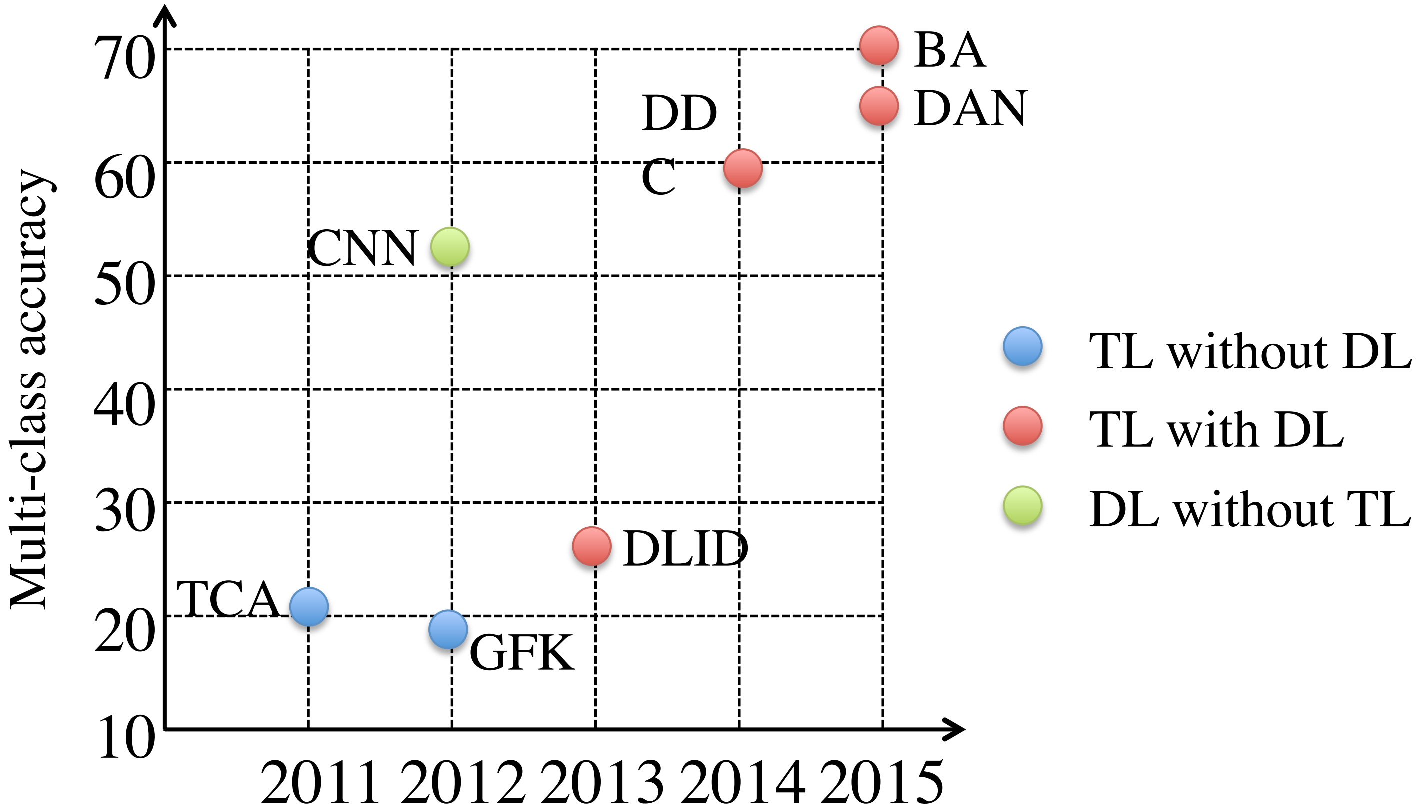
Benchmark Dataset: Office

- Description: leverage source images to improve classification of target images



Results

Unsupervised domain adaptation Amazon→Webcam over time



Applying Transfer Learning techniques outperforms directly applying Deep Learning models trained on the source.

With Deep Learning, Transfer Learning improves.

Overview

[5] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. "**Domain adaptation for large-scale sentiment classification: A deep learning approach**." ICML. 2011.

[6] Chopra, Sumit, Suhrid Balakrishnan, and Raghuraman Gopalan. "**Dlid: Deep learning for domain adaptation by interpolating between domains.**" ICML. 2013.

[7] Tzeng, Eric, et al. "**Deep domain confusion: Maximizing for domain invariance**." arXiv preprint arXiv:1412.3474. 2014.

[8] Long, Mingsheng, and Jianmin Wang. "**Learning transferable features with deep adaptation networks**." arXiv preprint arXiv:1502.02791. 2015.

[9] Ganin, Yaroslav, and Victor Lempitsky. "**Unsupervised Domain Adaptation by Backpropagation**." ICML. 2015.

[1] Nguyen, Hien V., et al. "**Joint hierarchical domain adaptation and feature learning.**" PAMI. 2013.

[2] Oquab, Maxime, et al. "**Learning and transferring mid-level image representations using convolutional neural networks**." CVPR 2014.

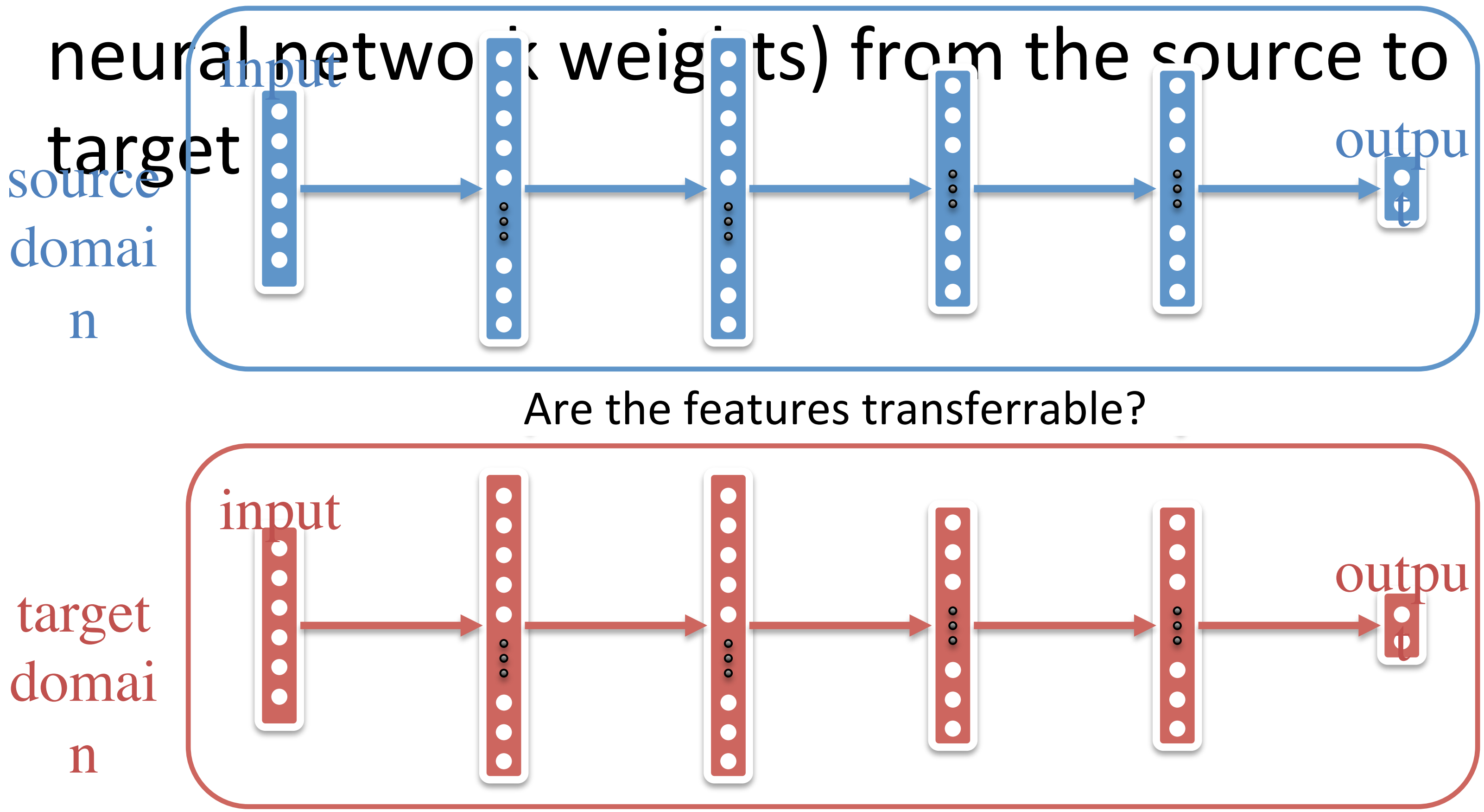
[3] Yosinski, Jason, et al. "**How transferable are features in deep neural networks**?." NIPS 2014.

[4] Tzeng, Eric, et al. "**Simultaneous deep transfer across domains and tasks**." CVPR. 2015.

data?

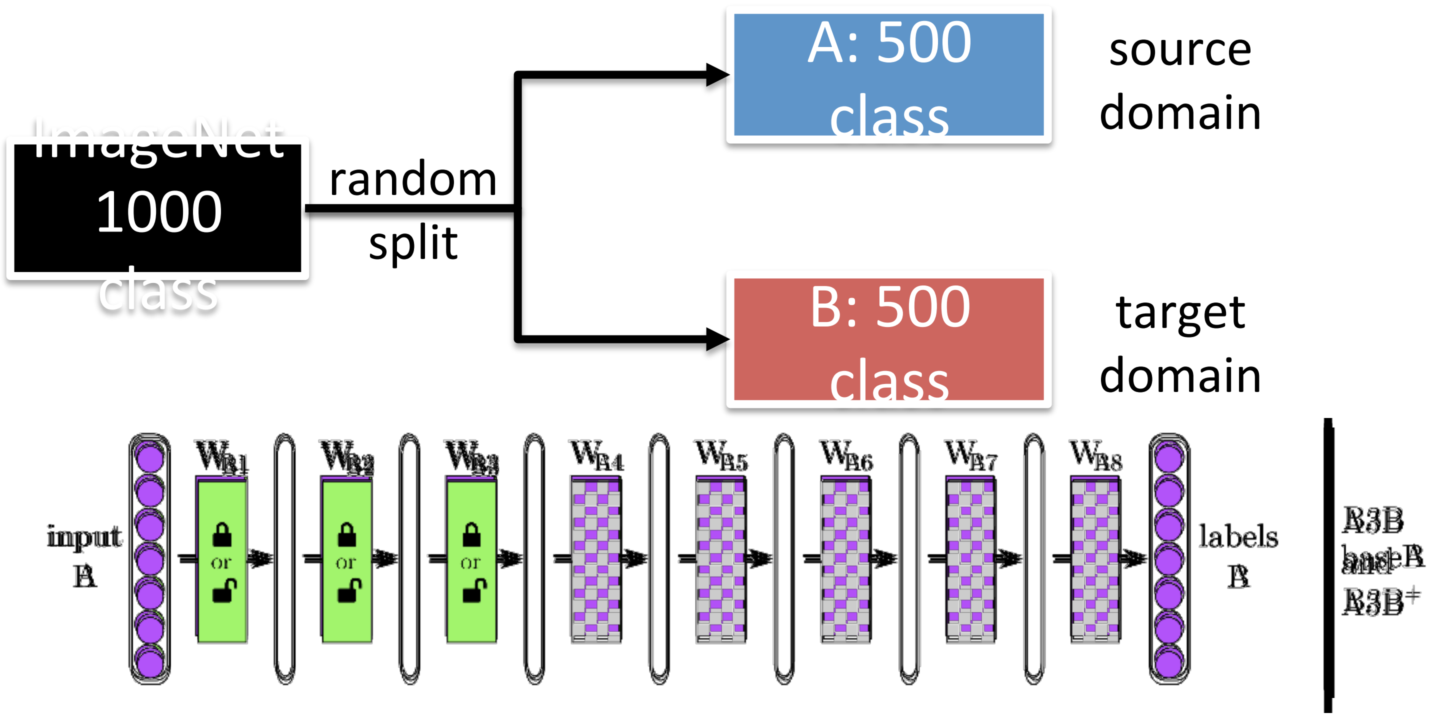
Single Modality

- Directly applying the model parameters (deep neural network weights) from the source to target



Single Modality

- Transferability of layer-wise features



AnB: initialize the first n layers with baseA and fix, randomly initialize the other layers and train with B

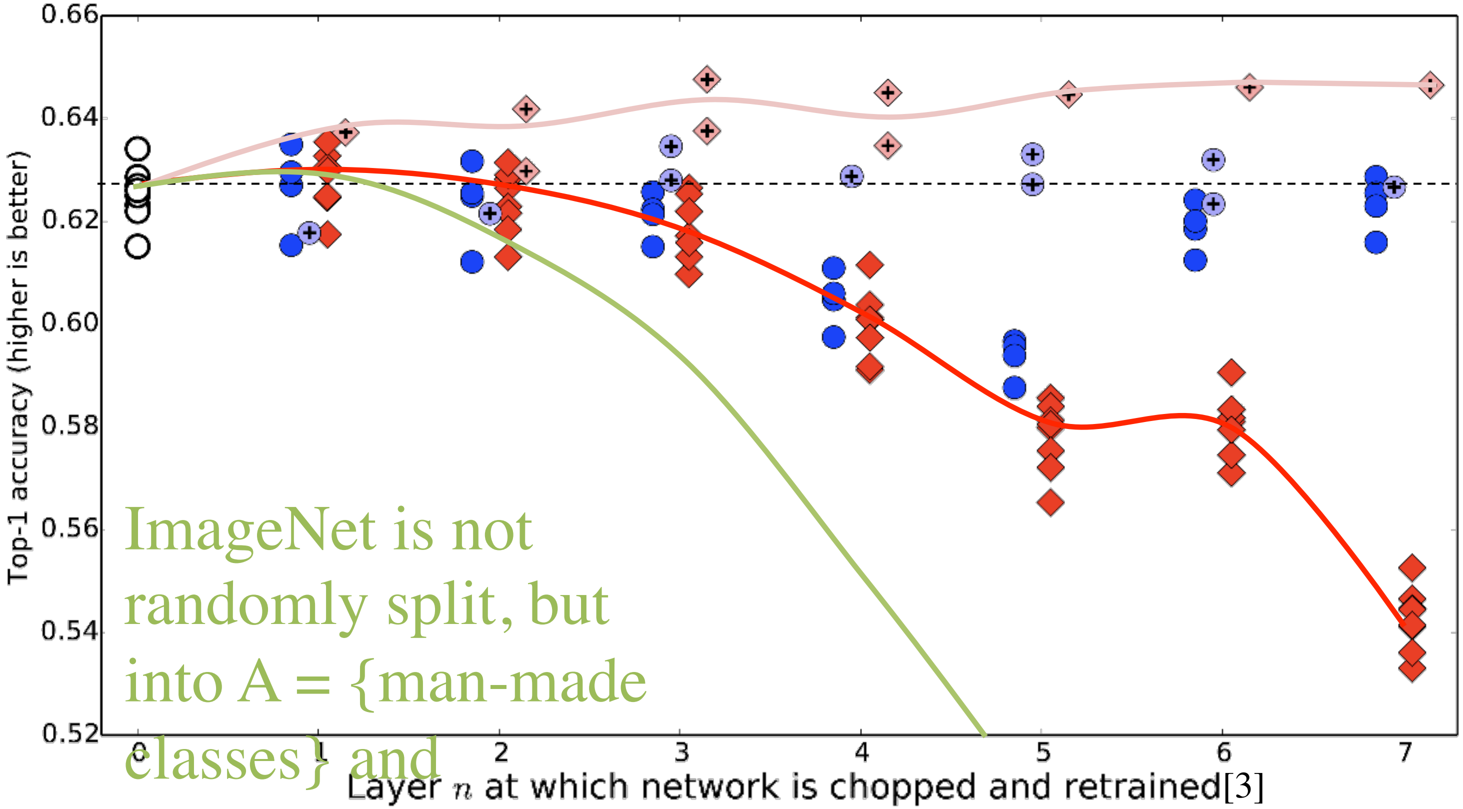
baseB: train all layers with B

baseA: train all layers with A

AnB+: initialize the first n layers with baseA, randomly initialize the other layers, and train all layers with B

Single Modality

- Transferability of layer-wise features



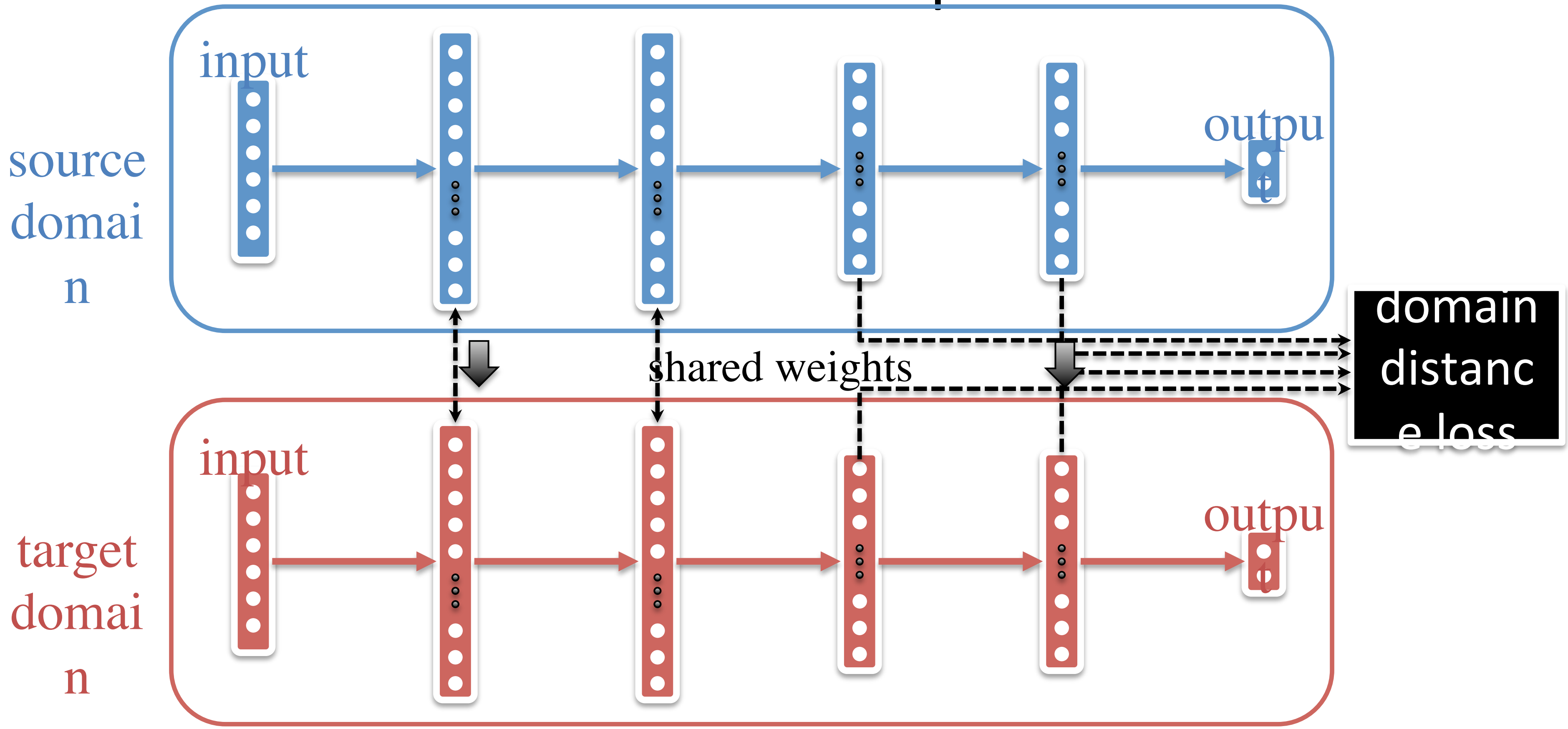
ImageNet is not randomly split, but into A = {man-made classes} and B = {natural classes}

Conclusion 1: lower layer features are more general and transferrable, and higher layer features are more specific and non-transferrable.

What happens if the source and target domain are very dissimilar?

Single Modality

- General framework of unsupervised transfer



For higher level features (more domain specific & not transferable), the source transfers to the target by minimizing domain distances.

For lower level features (more general & transferable), the source transfers to the target directly.

If some labelled target data are available, it would be better.

Single Modality

- Overall training objective

$$\mathcal{L} = \boxed{\mathcal{L}_C(X_S, y_S)} + \lambda \boxed{\mathcal{L}_D(X_S, X_T)}$$

source domain classification loss domain distance loss

- Domain distance losses
  - Maximum Mean Discrepancy [7]

$$MMD(X_S, X_T) = \| \frac{1}{\|X_S\|} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{\|X_T\|} \sum_{x_t \in X_T} \phi(x_t) \|_2^2$$

a particular representation, e.g. the representation after 5th layer

Single Modality

- Domain distance losses
  - MK-MMD (Multi-kernel variant of MMD) [8]

$$MK\!-\!MMD(X_S, X_T) = \| \frac{1}{\|X_S\|} \sum_{x_s \in X_S} \phi'(\phi(x_s)) - \frac{1}{\|X_T\|} \sum_{x_t \in X_T} \phi'(\phi(x_t)) \|_H^2$$

an embedding

$$k(\phi(x_s), \phi(x_t)) = \langle \phi'(\phi(x_s)), \phi'(\phi(x_t)) \rangle = \sum_{u=1}^{m} \beta_u k_u$$

$$\sum_{u=1}^{m} \beta_u = 1, \beta_u \geq 0, \forall u$$

Learn a more flexible distance metric than MMD by adjusting β_u

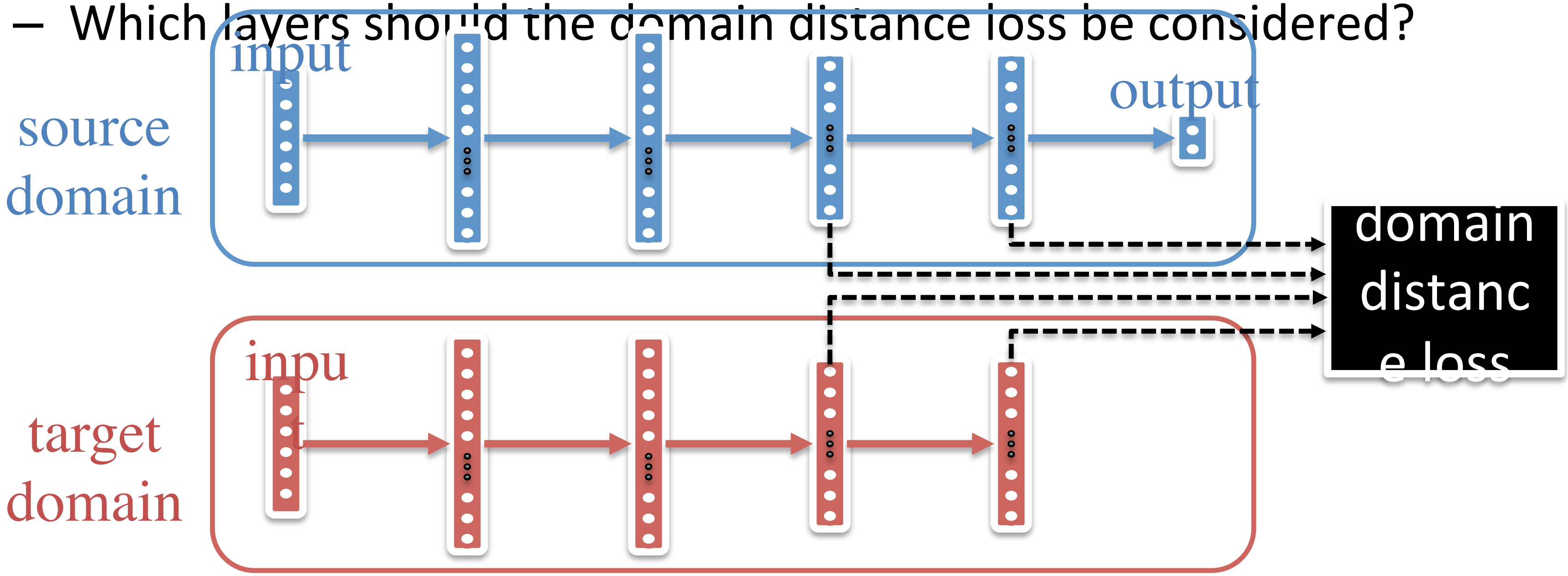
  - Domain classifier [4, 9]

$$\mathcal{L}_D(X_S, X_T) = -\sum_{i=1}^{\|X_S\|+\|X_T\|} \ell(x_i, d_i), \quad x_i \in X_s \cup X_t, \quad d_i = \begin{cases} 0 & x_i \in X_s \\ 1 & x_i \in X_t \end{cases}$$

A distribution-free metric - maximizes the domain classification error

Single Modality

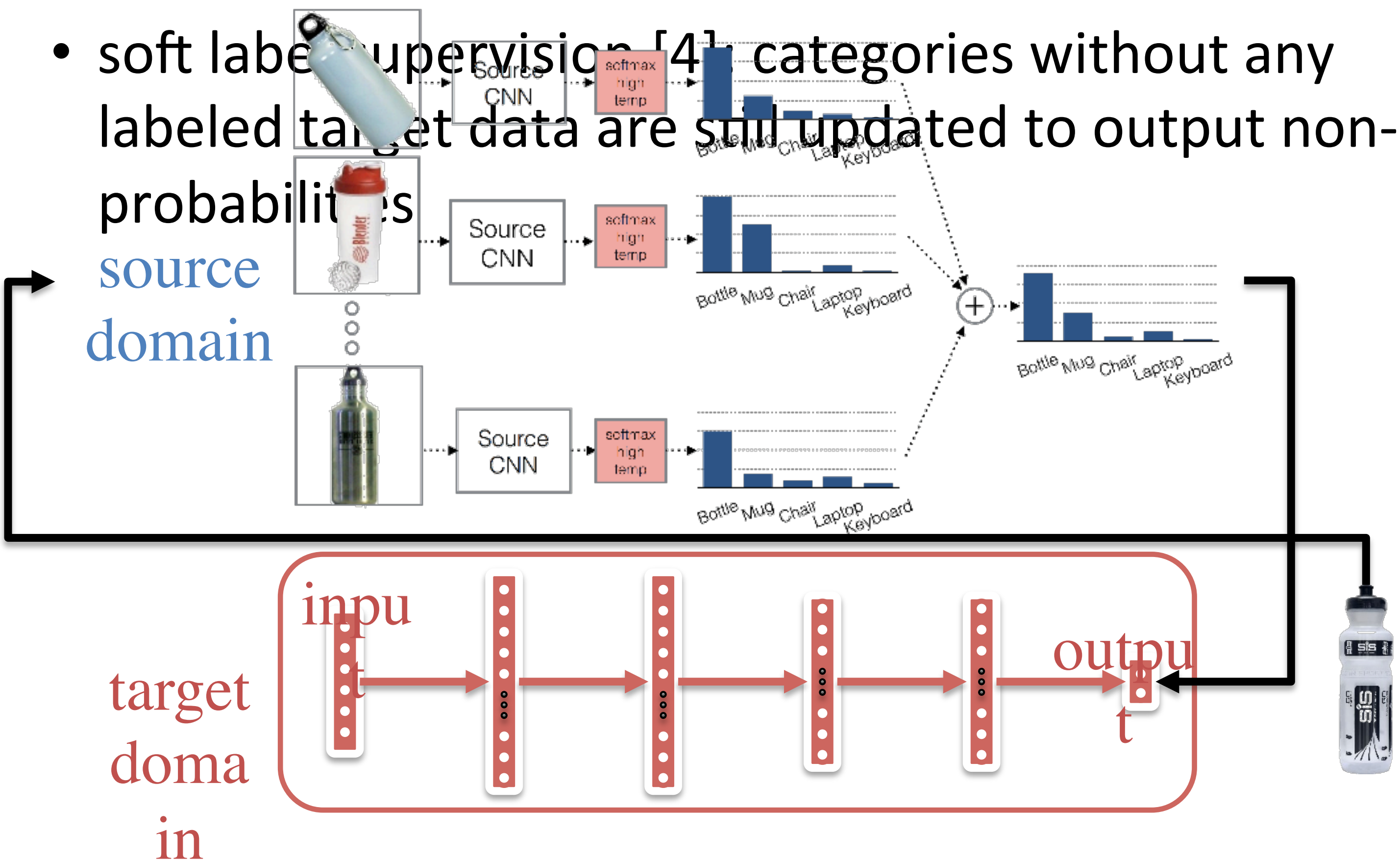
- Other factors to improve transfer
  - Which layers should the domain distance loss be considered?



- By learning, pinpoint the layer that minimizes the domain distance among all specific layers, say the fourth. [7]
- All the specific layers, say the last two layers. [8]

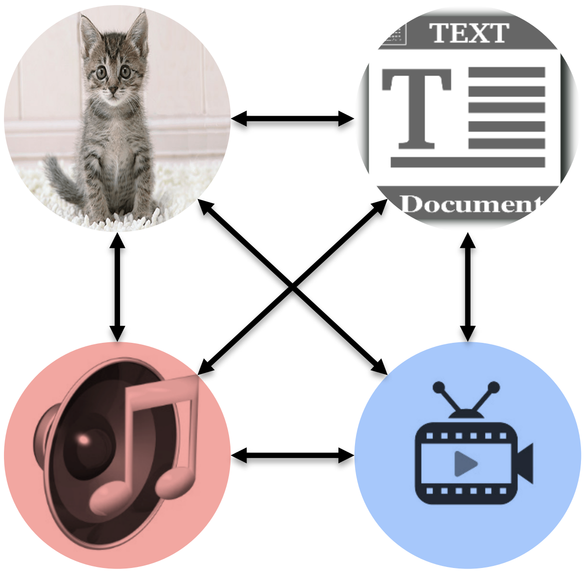
Single Modality

- Other factors to improve transfer
  - When we have some training data in the target domain?
    - soft label supervision [4]: categories without any labeled target data are still updated to output non-zero probabilities



Multiple Modalities

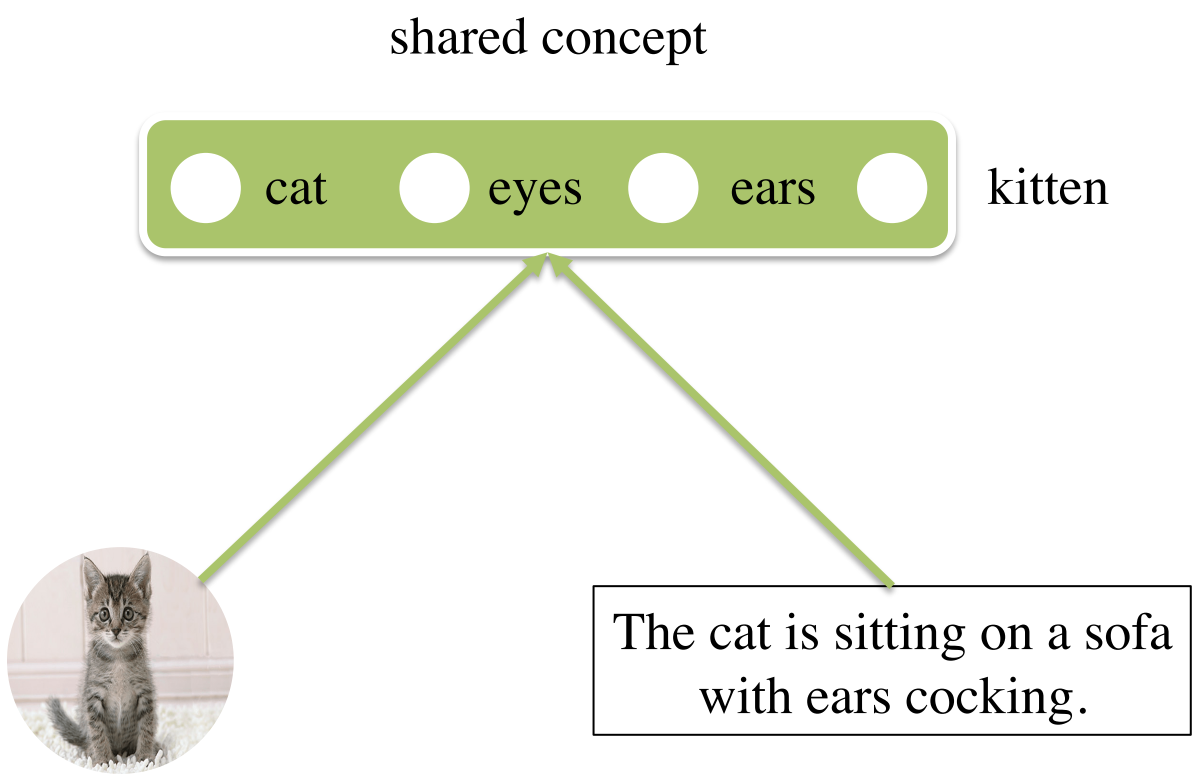
- The source domain and target domain could have different feature spaces, i.e., dimensionality.
  - Multimedia on the web
    - Images
    - Text documents
    - Audio
    - Video
  - Recommender systems
    - Douban
    - Taobao
    - Xiami Music
  - Robotics
    - Vision
    - Audio
    - Sensors



How to deal with multi-modal transfer with Deep Learning?

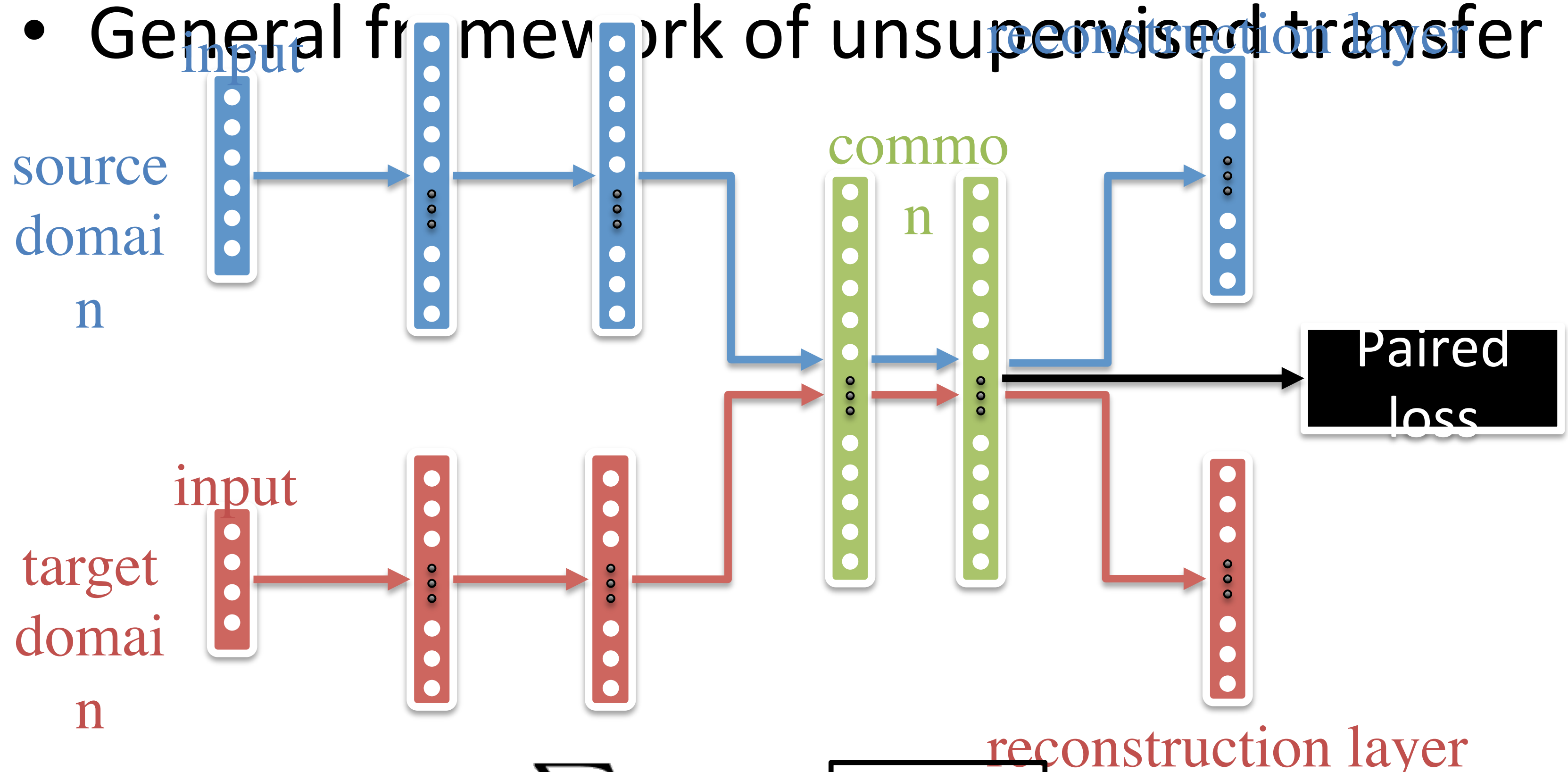
Multiple Modalities

- Key



Multiple Modalities

- General framework of unsupervised transfer



$$\sum_{x_s \in X_S, x_t \in X_T} \mathcal{L}_P(\boxed{\phi_C(x_s), \phi_C(x_t)})$$

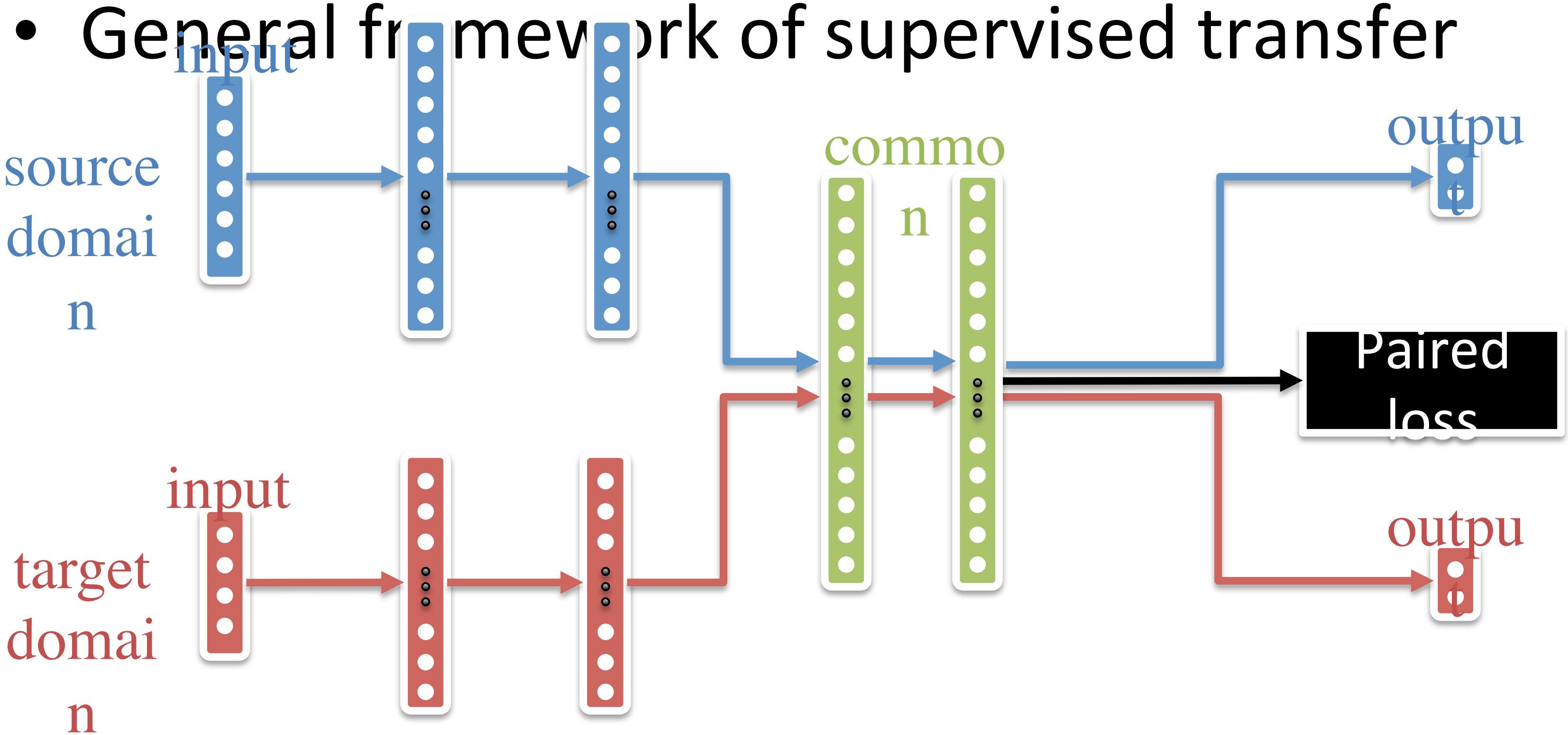
similarity

Paired loss: the similarity of a pair of source and target instances is preserved in the common space.

Reconstruction errors: $\mathcal{L}_R(X_S, \phi_R(X_S)) + \mathcal{L}_R(X_T, \phi_R(X_T))$

Multiple Modalities

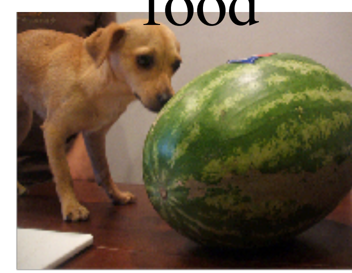
- General framework of supervised transfer



Classification loss: $\mathcal{L}_C(X_S, y_S) + \mathcal{L}_C(X_T, y_T)$

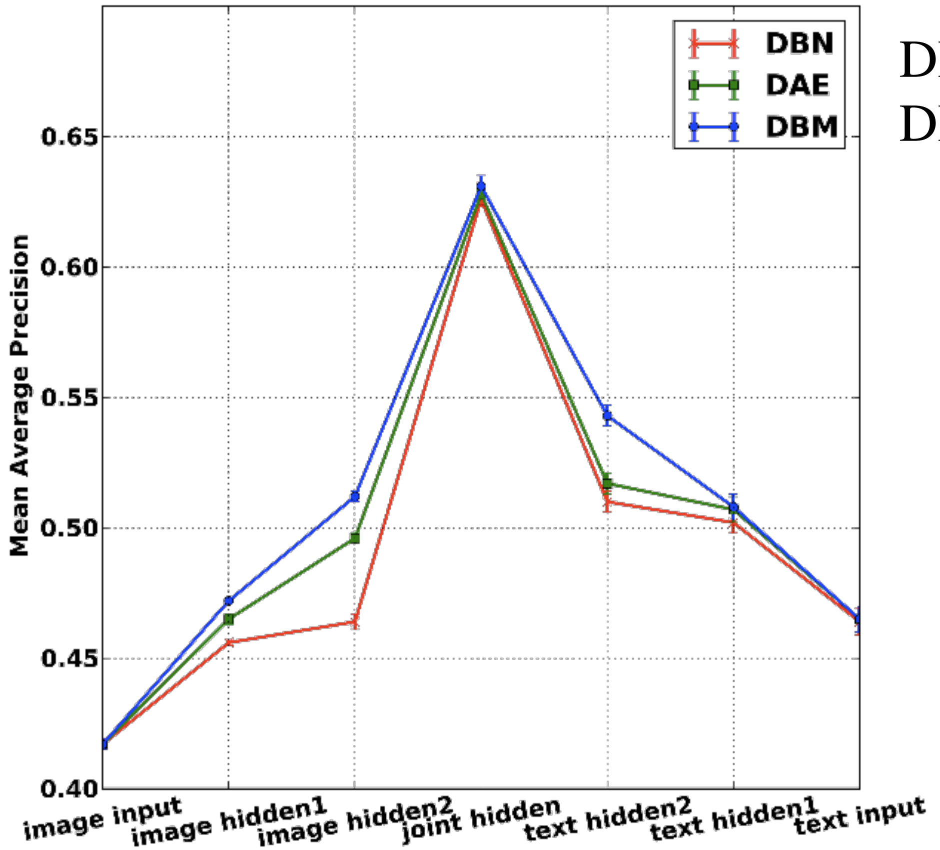
MIR-Flickr Dataset

- 1 million images with user-generated tags
  - 25,000 images are labelled with 24 categories
  - 10,000 for training, 5,000 for validation, 10,000 for testing

categories	baby, female, portrait, people	plant life, river, water	clouds, sea, sky, transport, water	animals, dog, food
domain 1: images				
domain 2: text	claudia	⟨ no text ⟩	barco, pesca, boattosail, navegação	watermelon, hilarious, chihuahua, dog

Results

Mean Average Precision (MAP) by applying LR to different layers [13]



DBN [12]

DBM [13]

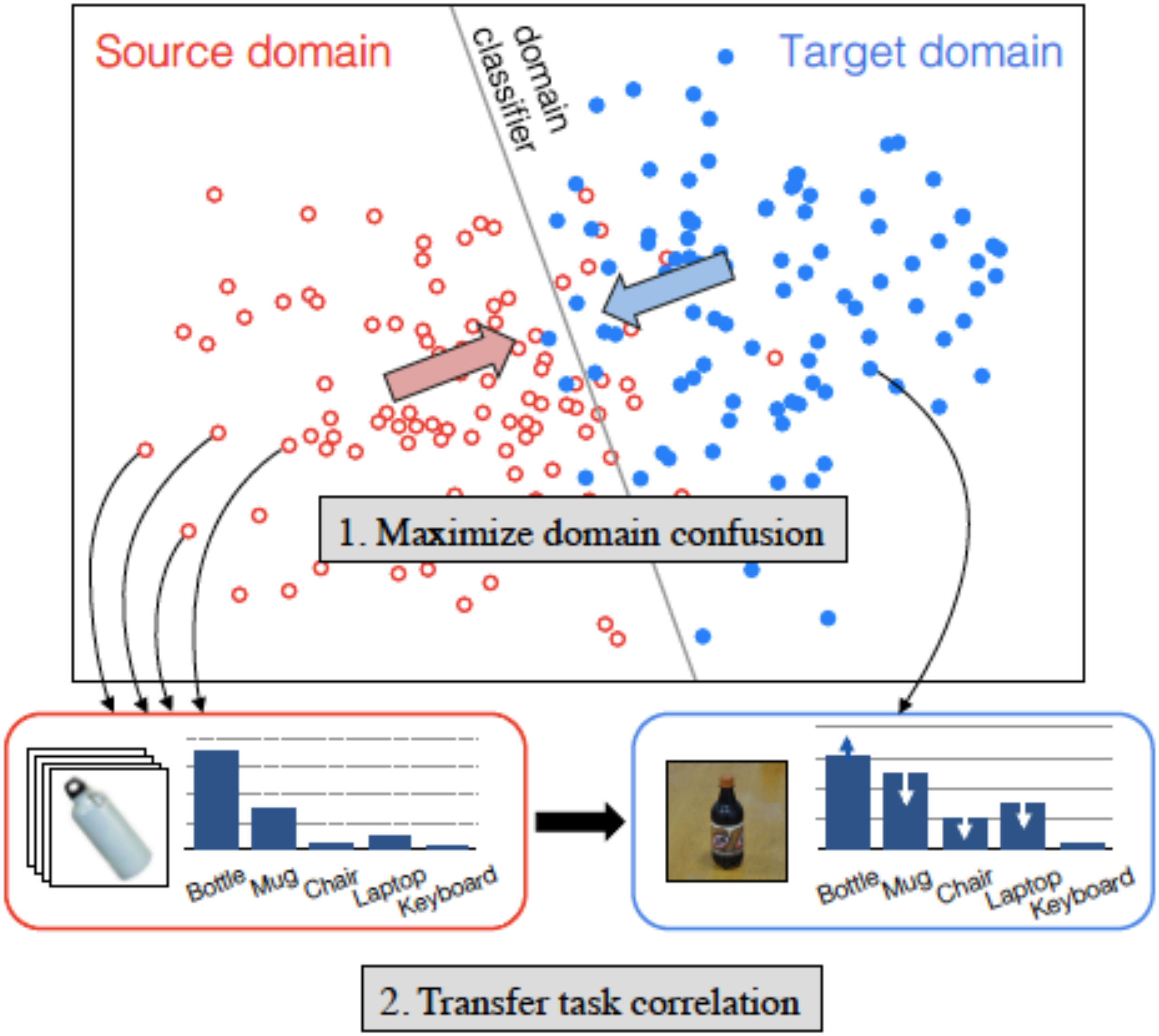
Transferring either one of the two domains to the other (joint hidden), outperforms the domain itself (image_input OR text_input).

References

[1] Nguyen, Hien V., et al. "**Joint hierarchical domain adaptation and feature learning.**" PAMI. 2013.
[2] Oquab, Maxime, et al. "**Learning and transferring mid-level image representations using convolutional neural networks**." CVPR. 2014.
[3] Yosinski, Jason, et al. "**How transferable are features in deep neural networks**?." NIPS. 2014.
[4] Tzeng, Eric, et al. "**Simultaneous deep transfer across domains and tasks**." CVPR. 2015.
[5] Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. "**Domain adaptation for large-scale sentiment classification: A deep learning approach**." ICML. 2011.
[6] Chopra, Sumit, Suhrid Balakrishnan, and Raghuraman Gopalan. "**Dlid: Deep learning for domain adaptation by interpolating between domains.**" ICML. 2013.
[7] Tzeng, Eric, et al. "**Deep domain confusion: Maximizing for domain invariance**." arXiv preprint arXiv:1412.3474. 2014.
[8] Long, Mingsheng, and Jianmin Wang. "**Learning transferable features with deep adaptation networks**." arXiv preprint arXiv:1502.02791. 2015.
[9] Ganin, Yaroslav, and Victor Lempitsky. "**Unsupervised Domain Adaptation by Backpropagation**." ICML. 2015.
[10] Huang, Jui-Ting, et al. "**Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers**." ICASSP. 2013.
[11] Gupta, Saurabh, Judy Hoffman, and Jitendra Malik. "**Cross Modal Distillation for Supervision Transfer**." arXiv preprint arXiv:1507.00448. 2015.
[12] Ngiam, Jiquan, et al. "**Multimodal deep learning**." ICML. 2011.
[13] Srivastava, Nitish, and Ruslan Salakhutdinov. "**Multimodal learning with deep Boltzmann machines**." JMLR. 2014
[14] Sohn, Kihyuk, Wenling Shang, and Honglak Lee. "**Improved multimodal deep learning with variation of information**." NIPS. 2014.

Simultaneous Deep Transfer Across Domains and Tasks

Eric Tzeng, Judy Hoffman, Trevor Darrell, Kate Saenko, ICCV 2015



Tzeng et al.: Architecture

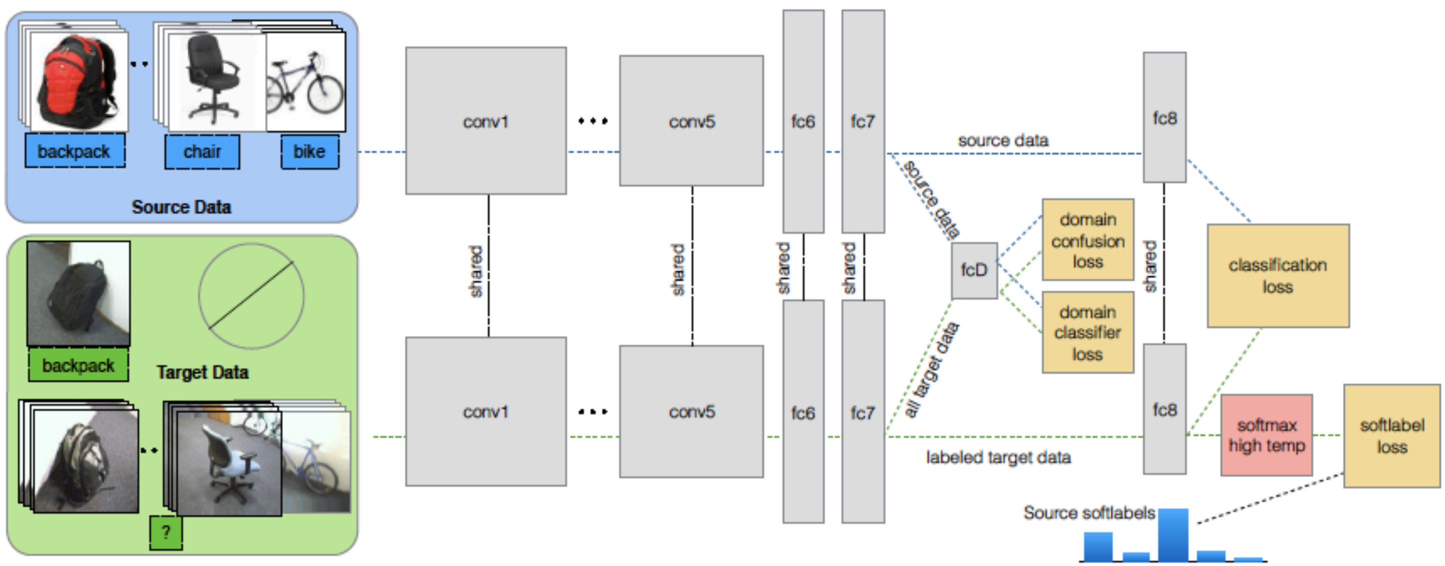


Figure 2. Our overall CNN architecture for domain and task transfer. We use a domain confusion loss over all source and target (both labeled and unlabeled) data to learn a domain invariant representation. We simultaneously transfer the learned source semantic structure to the target domain by optimizing the network to produce activation distributions that match those learned for source data in the source only CNN. *Best viewed in color.*

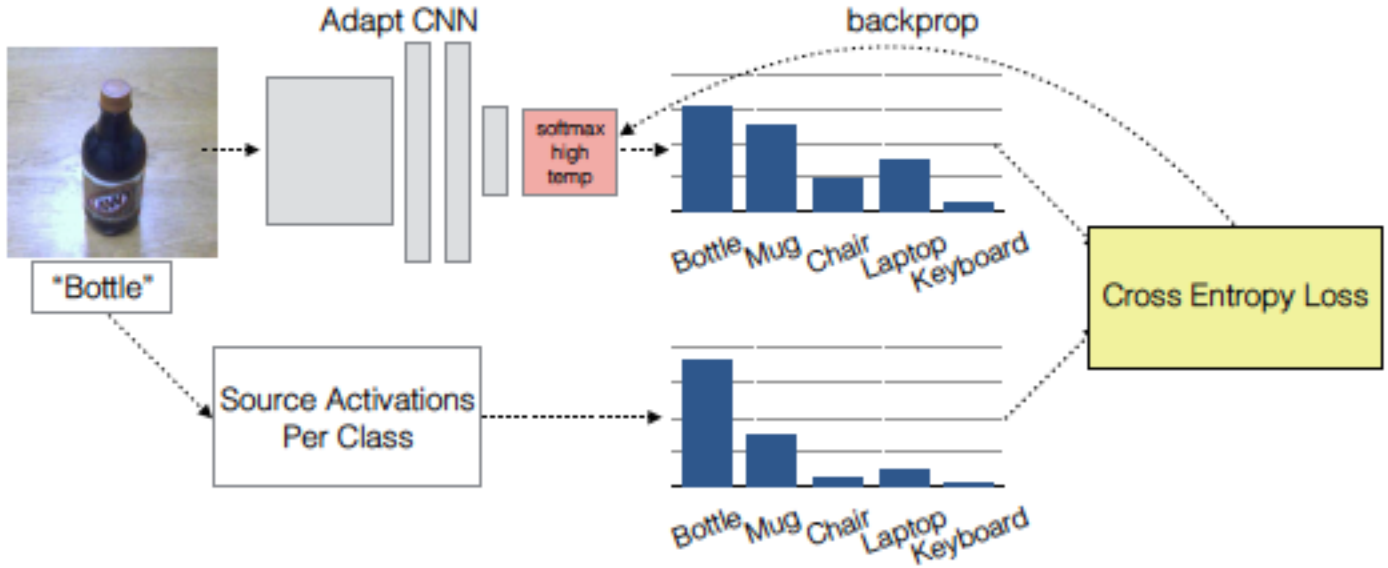


Figure 4. Depiction of the use of source per-category soft activations with the cross entropy loss function over the current target activations.

$$\begin{aligned}
\mathcal{L}(x_S, y_S, x_T, y_T, \theta_D; &\theta_{\text{repr}}, \theta_C) = \\
&\mathcal{L}_C(x_S, y_S, x_T, y_T; \theta_{\text{repr}}, \theta_C) \\
&+ \lambda \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}) \\
&+ \nu \mathcal{L}_{\text{soft}}(x_T, y_T; \theta_{\text{repr}}, \theta_C).
\end{aligned}$$

Tzeng et al.: Architecture

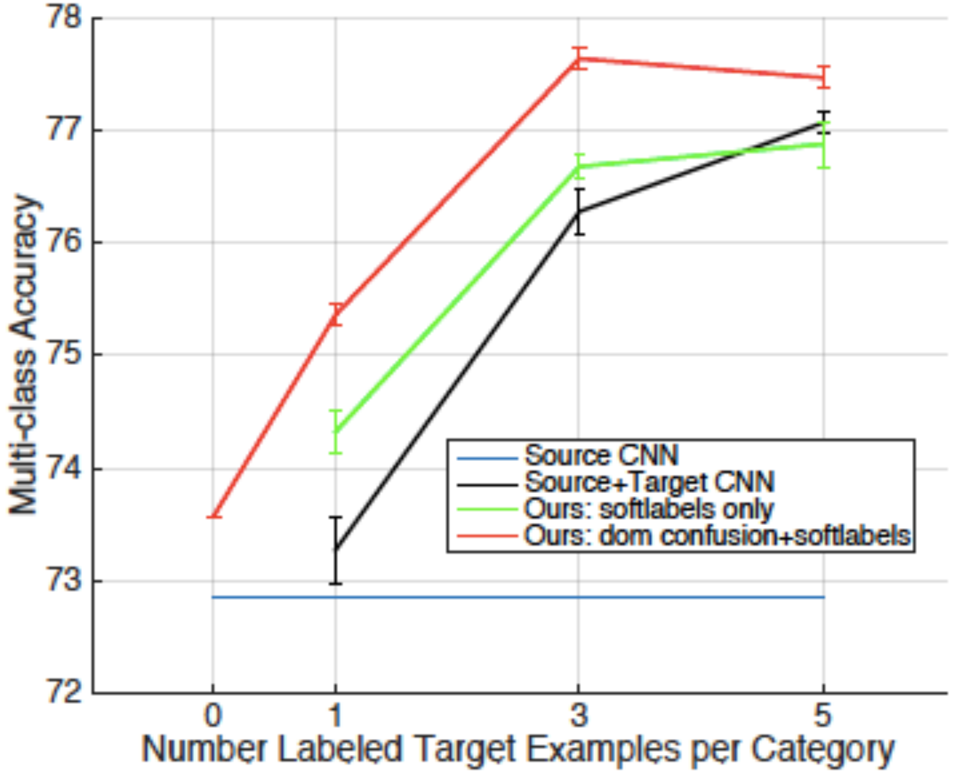
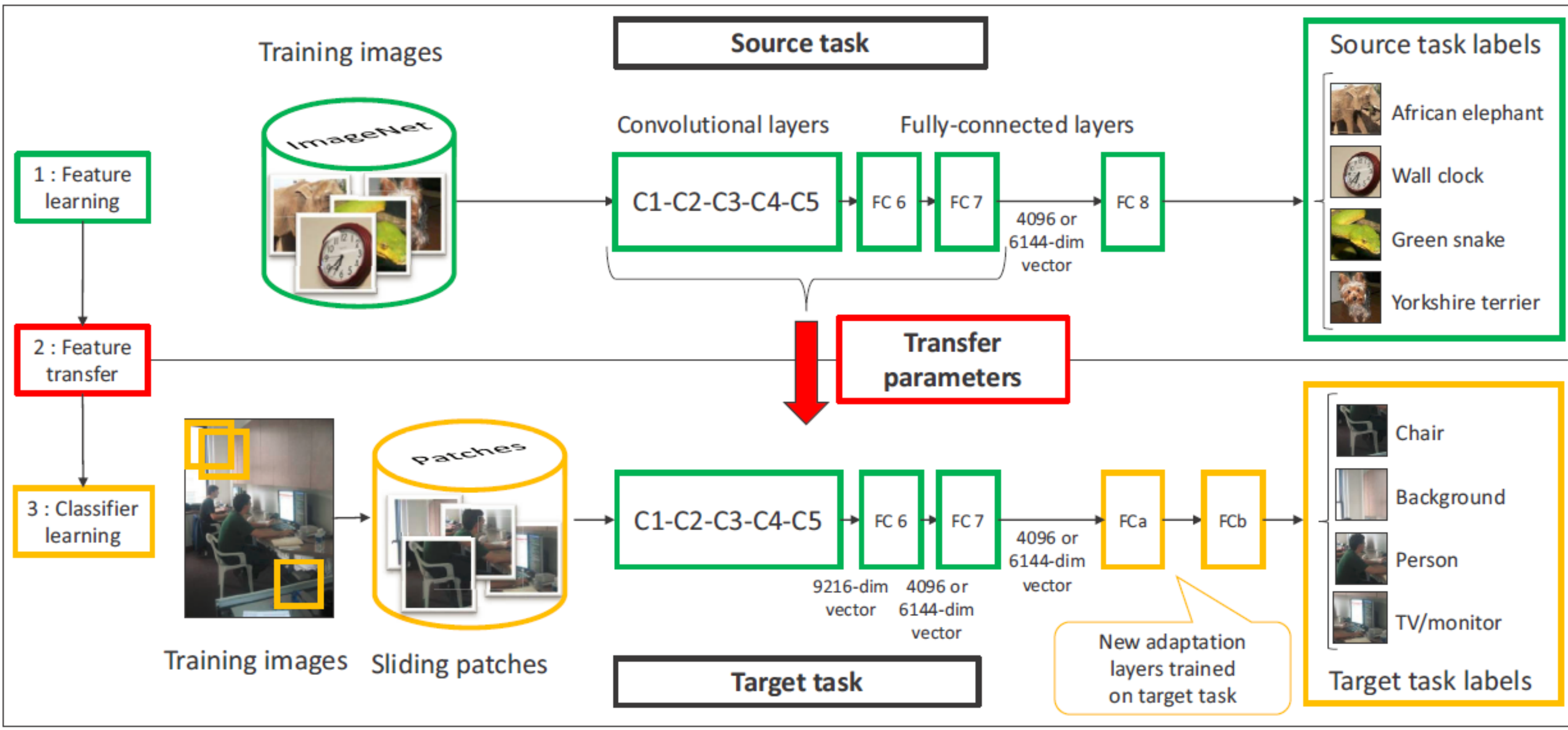


Figure 6. ImageNet→Caltech supervised adaptation from the Cross-dataset [30] testbed with varying numbers of labeled target examples per category. We find that our method using soft label loss

Oquab, Bottou, Laptev, Sivic: Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. CVPR 2014.



Transfer Learning in Convolutional Neural Networks

- Source Domain: ImageNet
  - 1000 classes, 1.2 million images
- Target Domain: Pascal VOC 2007 object classification
  - 20 classes, about 5000 images
- PRE-1000C: the proposed method

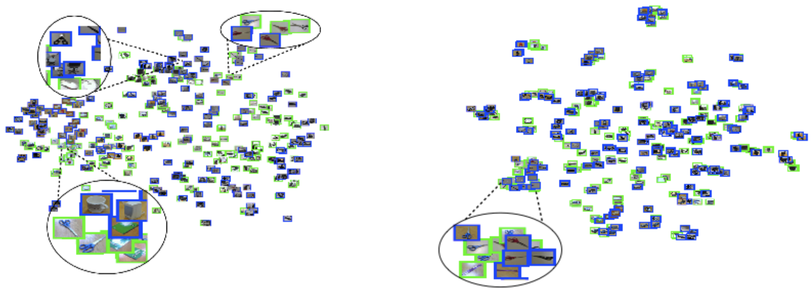
	plane	bike	bird	boat	btl	bus	car	cat	chair	cow	table	dog	horse	moto	pers	plant	sheep	sofa	train	tv	**mAP**
INRIA [33]	77.5	63.6	56.1	71.9	33.1	60.6	78.0	58.8	53.5	42.6	54.9	45.8	77.5	64.0	85.9	36.3	44.7	50.6	79.2	53.2	59.4
NUS-PSL [46]	82.5	79.6	64.8	73.4	**54.2**	75.0	77.5	79.2	46.2	62.7	41.4	74.6	**85.0**	76.8	91.1	53.9	61.0	**67.5**	83.6	70.6	70.5
PRE-1000C	**88.5**	**81.5**	**87.9**	**82.0**	47.5	**75.5**	**90.1**	**87.2**	**61.6**	**75.7**	**67.3**	**85.5**	83.5	**80.0**	**95.6**	**60.8**	**76.8**	58.0	**90.4**	**77.9**	**77.7**

Per-class results for object classification on the VOC2007 test set (average precision %)

DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition

- Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, Trevor Darrell. ICML2014
- Questions:
  - How to transfer features to tasks with different labels
  - Do features extracted from the CNN generalize to other datasets?
  - How does performance vary with network depth?
- Algorithm:
  - A deep convolutional model is first trained in a fully supervised setting using a state-of-the-art method Krizhevsky et al. (2012).
  - extract various features from this network, and evaluate the efficacy of these features on generic vision tasks.

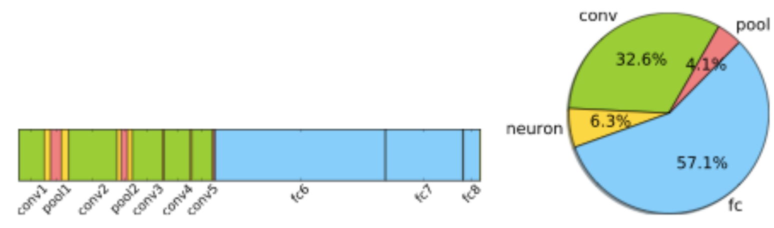
Comparison: DECAF to others



(a) SURF features

(b) $DeCAF_6$

ure 5. Visualization of the webcam (green) and dslr (blue) domains using the original released SURF features (a) and $DeCAF_6$ (l
e figure is best viewed by zooming in to see the images in local regions. All images from the scissor class are shown enlarged. Th
well clustered and overlapping in both domains with our representation, while SURF only clusters a subset and places the others
joint parts of the space, closest to distinctly different categories such as chairs and mugs.

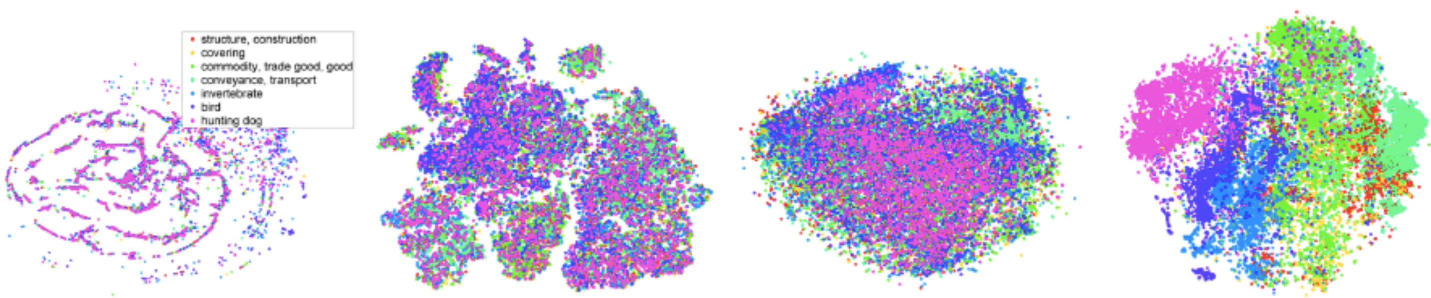


(a)

(b)

Figure 3. (a) The computation time on each layer when running classification on one single input image. The layers with the most time consumption are labeled. (b) The distribution of computation

DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition



(a) LLC

(b) GIST

(c) $DeCAF_1$

(d) $DeCAF_6$

Figure 1. This figure shows several t-SNE feature visualizations on the ILSVRC-2012 validation set. (a) LLC , (b) GIST, and features derived from our CNN: (c) $DeCAF_1$, the first pooling layer, and (d) $DeCAF_6$, the second to last hidden layer (best viewed in color).

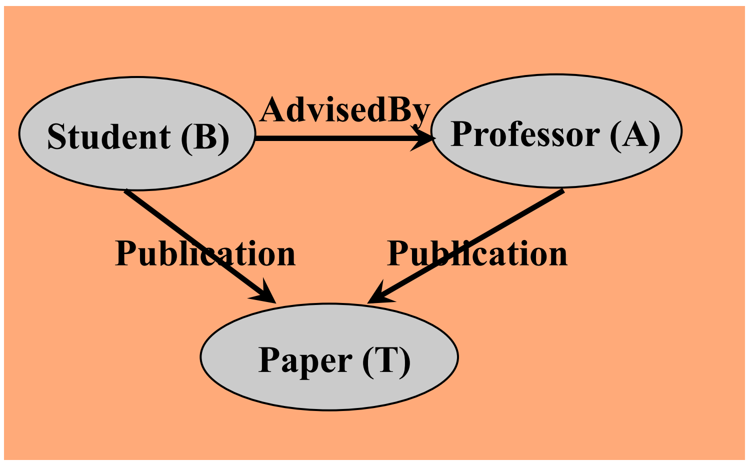
Relational Transfer Learning Approaches

- **Motivation:**
  - If two logically described domains (relational, data is non-i.i.d) are related, they must share ***similar relations*** among objects.
  - These relations can be used for transfer learning

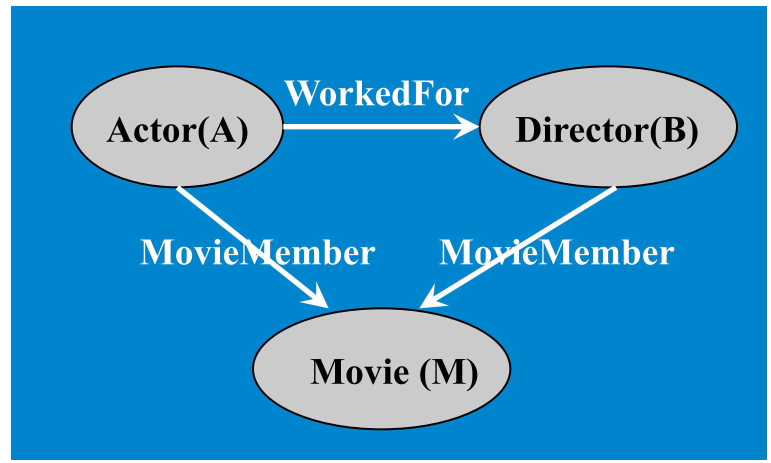
Relational Transfer Learning Approaches (cont.)

[Mihalkova *etal*., AAAI-07, Davis and Domingos, ICML-09]

Academic domain (source)



Movie domain (target)



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

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

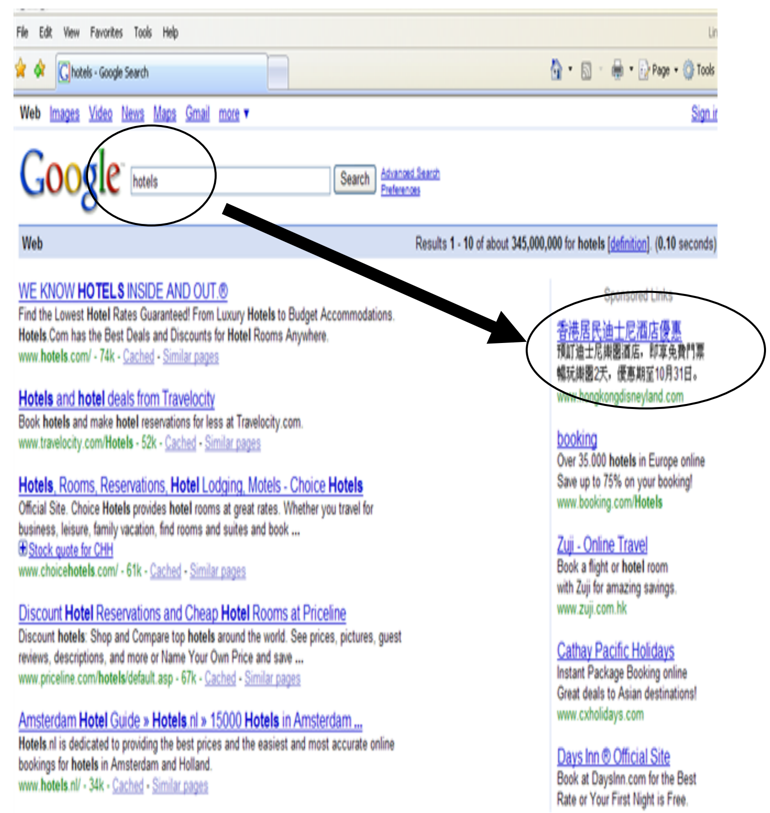
P1(x, y) ∧ P2 (x, z) => P2 (y, z)

迁移学习应用

TRANSFER LEARNING APPLICATIONS

Query Classification and Online Advertisement

- ACM KDDCUP 05 Winner
- SIGIR 06
- ACM Transactions on Information Systems Journal 2006
  - Joint work with Dou Shen, Jiantao Sun and Zheng Chen



QC as Machine Learning

Inspired by the KDDCUP'05 competition

– Clas

– Que

– Targ each

nod

Table I. Examples of queries.

1967 shelby mustang
actress hildegarde
Aldactone
alfred Hitchcock
amazon rainforest
section8rentalhouses.com
Sakpsabancnnhayat

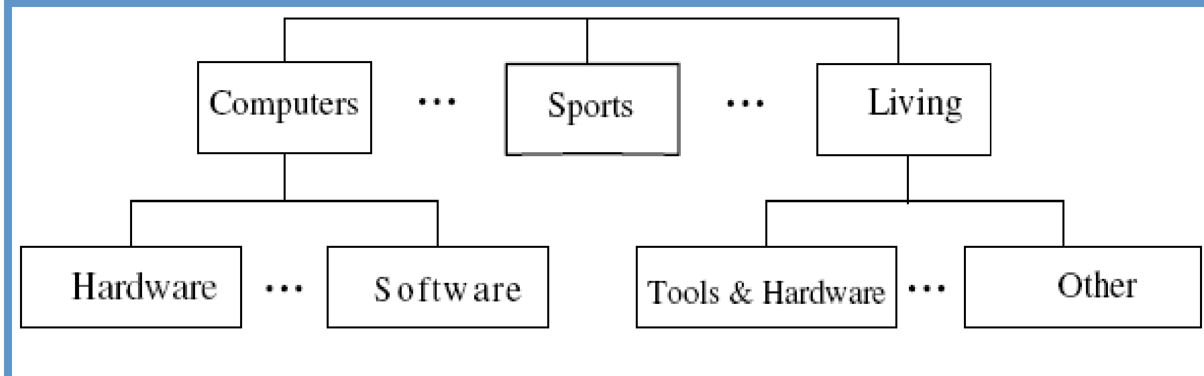
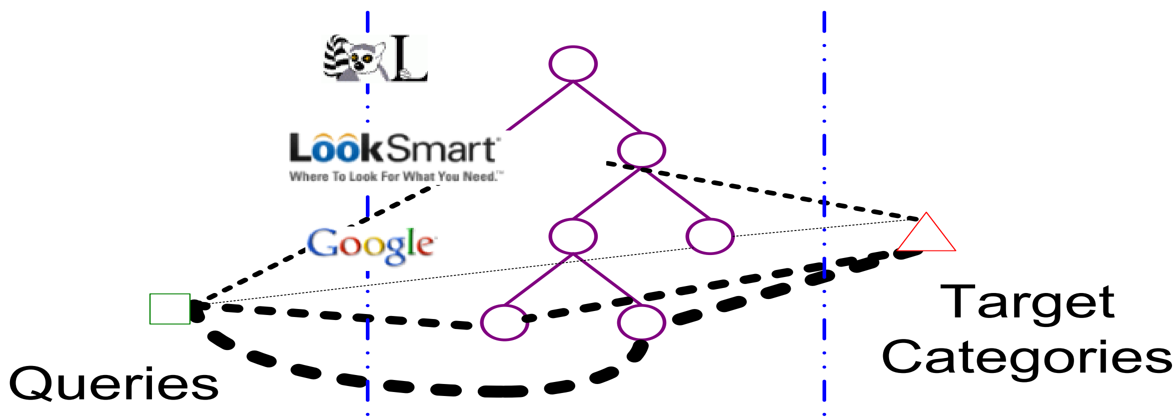


Figure 1: An Example of the Target Taxonomy.

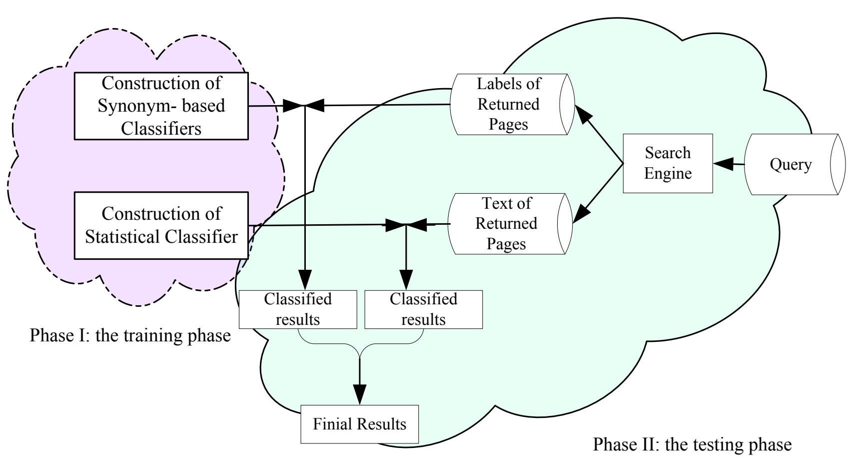
Target-transfer Learning in QC

- Classifier, once trained, stays constant
  - Target Classes Before
    - Sports, Politics (European, US, China)
  - Target Classes Now
    - Sports (Olympics, Football, NBA), Stock Market (Asian, Dow, Nasdaq), History (Chinese, World) How to allow target to change?
- Application:
  - advertisements come and go,
  - but our query→target mapping needs not be retrained!
- We call this the target-transfer learning problem

Solutions: Query Enrichment + Staged Classification

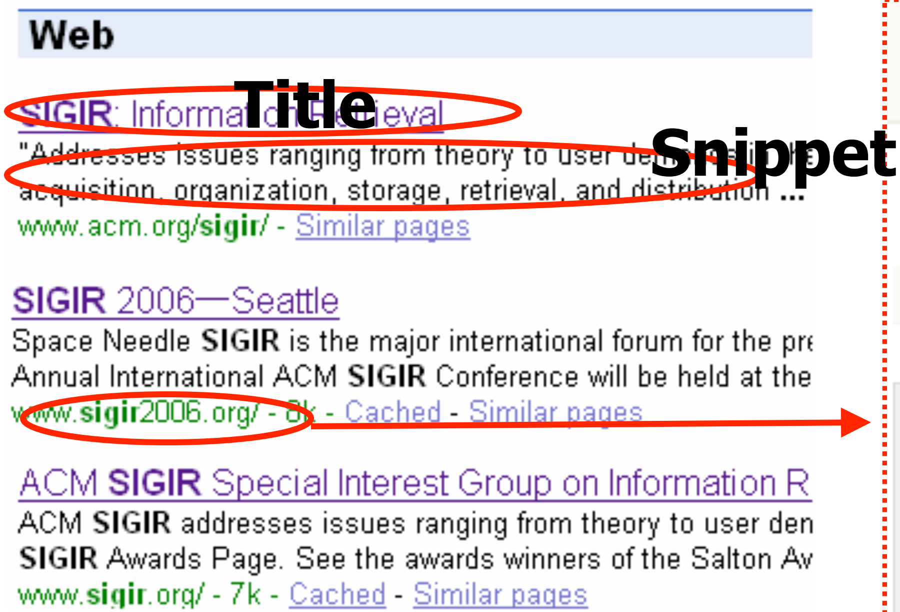


Solution: Bridging classifier



Step 1: Query enrichment

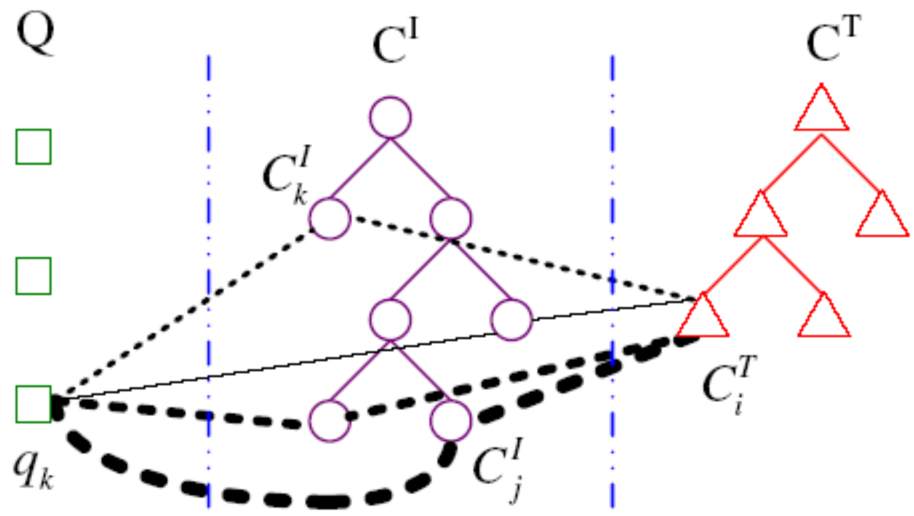
- Textual information
- Category information



Full text

Step 2: Bridging Classifier

- Wish to avoid:
  - When target is changed, training needs to repeat!
- Solution:
  - Connect the target taxonomy and queries by taking an intermediate taxonomy as a bridge



Bridging Classifier (Cont.)

- How to connect?

$$
\begin{aligned}
p(C_i^T|q) &= \sum_{C_j^I} p(C_i^T, C_j^I|q) \\
&= \sum_{C_j^I} p(C_i^T|C_j^I, q)p(C_j^I|q) \\
&\approx \sum_{C_j^I} p(C_i^T|C_j^I)p(C_j^I|q) \\
&= \sum_{C_j^I} p(C_i^T|C_j^I)\frac{p(q|C_j^I)p(C_j^I)}{p(q)} \\
&\propto \sum_{C_j^I} p(C_i^T|C_j^I)p(q|C_j^I)p(C_j^I)
\end{aligned}
$$

The relation between C_i^T and C_j^I

The relation between q and C_j^I

Prior prob. of C_j^I

The relation between q and C_i^T

$$\boxed{c^* = \arg\max_{C_i^T} p(C_i^T|q)}$$

Category Selection for Intermediate Taxonomy

– Category Selection for Reducing Complexity

- Total Probability (TP)

$$Score(C_j^I) = \sum_{C_i^T} \hat{P}(C_i^T | C_j^I)$$

- Mutual Information

$$MI(C_i^T, C_j^I) = \frac{1}{|C_i^T|} \sum_{t \in C_i^T} MI(t, C_j^I)$$

$$MI_{avg}(C_j^I) = \sum_{C_j^T} MI(C_i^T, C_j^I)$$

Result of Bridging Classifiers

- Performance of the Bridging Classifier with Different Granularity of Intermediate Taxonomy

	Top 2	Top 3	Top 4	Top 5	Top All
F1	0.267	0.285	0.312	0.352	**0.424**
Precision	0.270	0.291	0.339	0.368	**0.447**

– Using bridging classifier allows the target classes to change freely

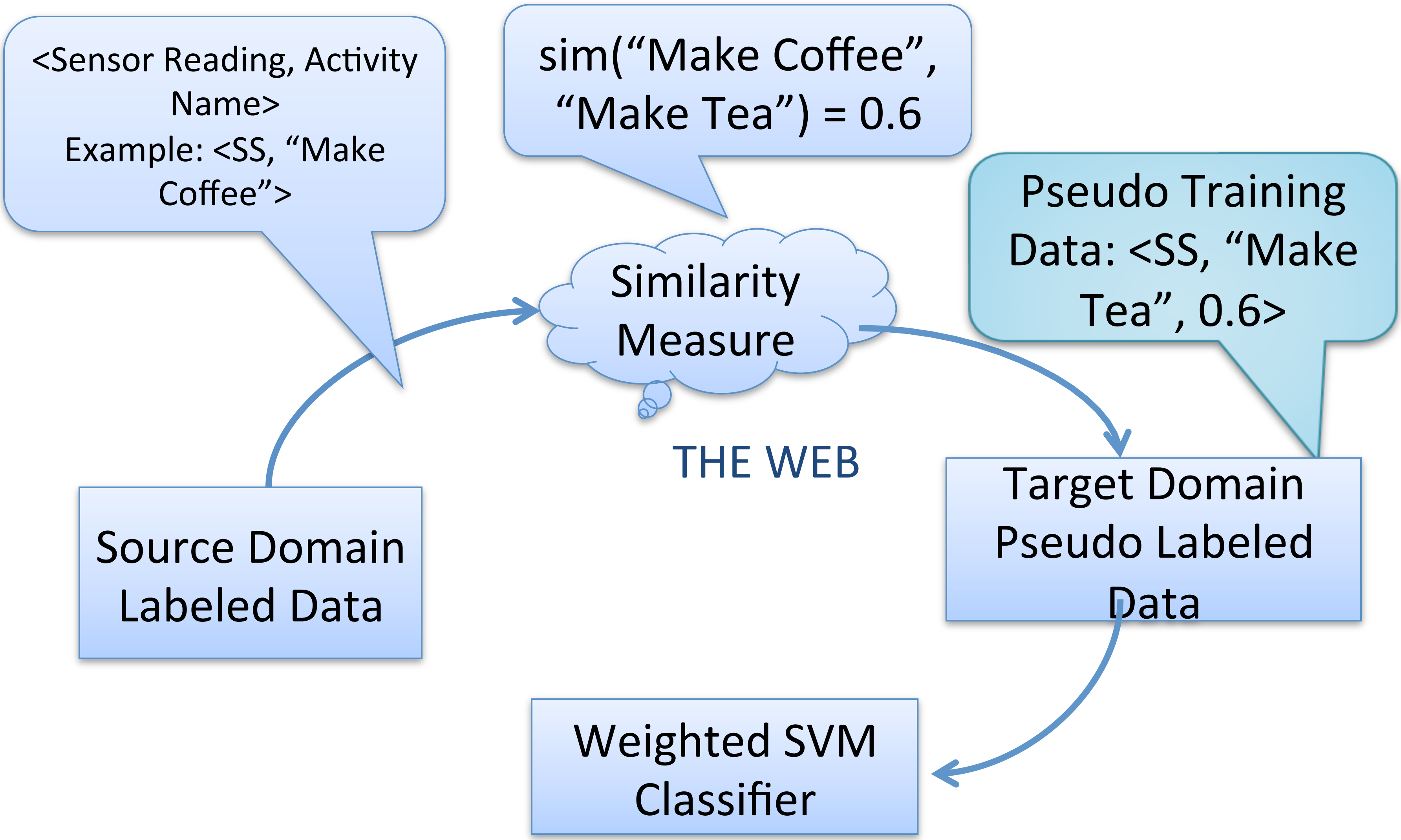
- no the need to retrain the classifier!

Cross Domain Activity Recognition
[Zheng, Hu, Yang, Ubicomp 2009]

- Challenges:
  - A new domain of activities without labeled data
- Cross-domain activity recognition
  - Transfer some available labeled data from source activities to help training the recognizer for the target activities.

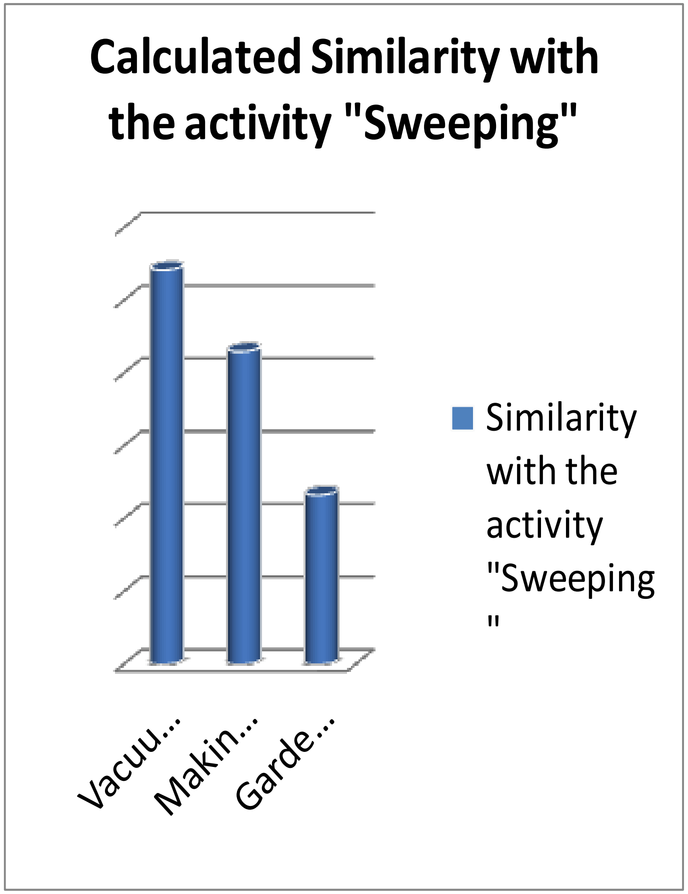
Dishwashing

How to use the similarities?



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”



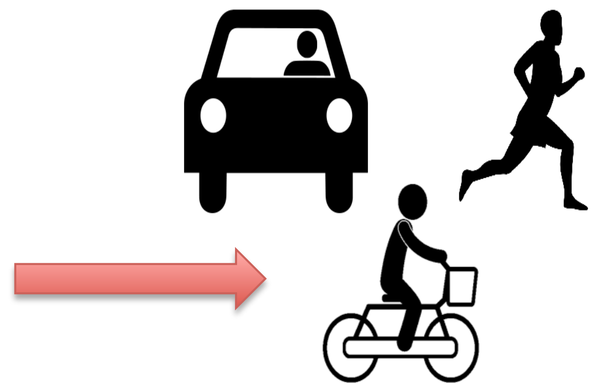
Cross-Domain AR: Performance

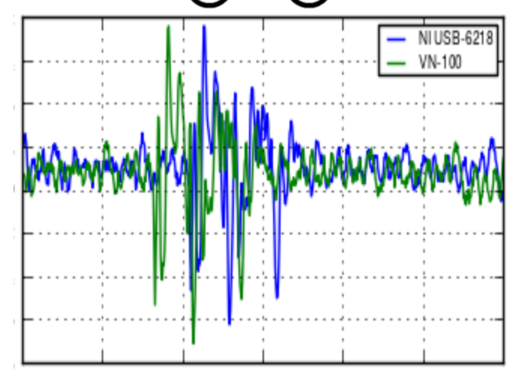
	Mean Accuracy with Cross Domain Transfer	# Activities (Source Domain)	# Activities (Target Domain)	Baseline (Random Guess)
MIT Dataset (Cleaning to Laundry)	58.9%	13	8	12.5%
MIT Dataset (Cleaning to Dishwashing)	53.2%	13	7	14.3%
Intel Research Lab Dataset	63.2%	5	6	16.7%

- Activities in the source domain and the target domain are generated from ten random trials, mean accuracies are reported.

Transferring knowledge from social to physical

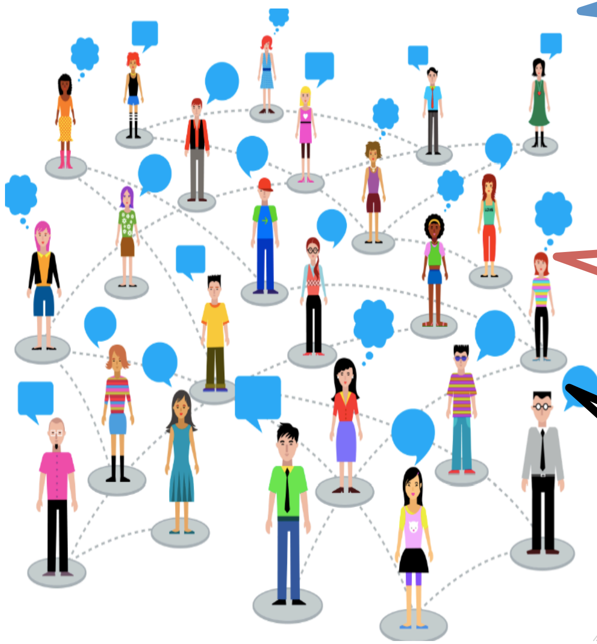
- Ubiquitous physical sensors motivate extensive research on ubiquitous computing.





Which activity is this person performing?

Transferring from social to physical



I am on a business trip in New York. The Metropolitan Museum of Art is fantastic!

Brilliant night at Chilli Food, wine, hospitality all excellent. Bristol's top restaurant.

Back in the #gym after 3.5 weeks :) feeling good #exercise

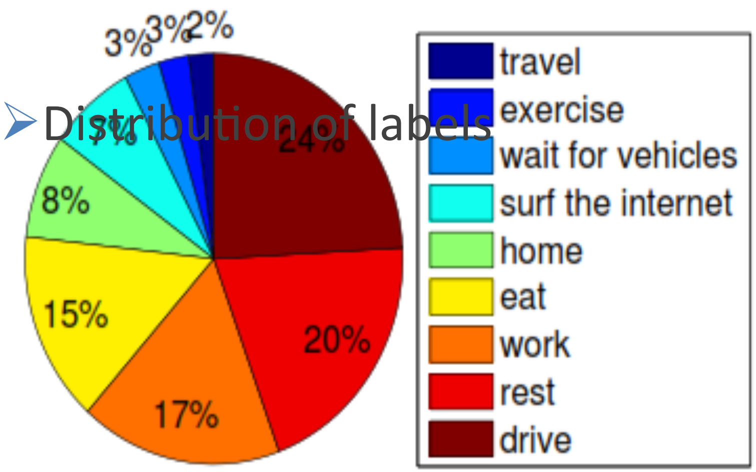
Can we transfer knowledge from social media to physical world?

Transfer from social to physical

Cellphone Sensor Dataset

- 232 sensor records
- 10 volunteers

- Distribution of labels

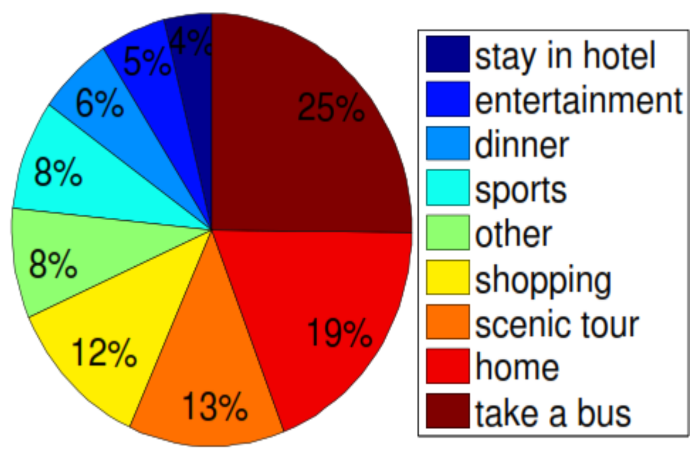


ion

Sina Weibo

- Distribution of top 9 labels

- 10,791 tweets



Transfer from social to physical

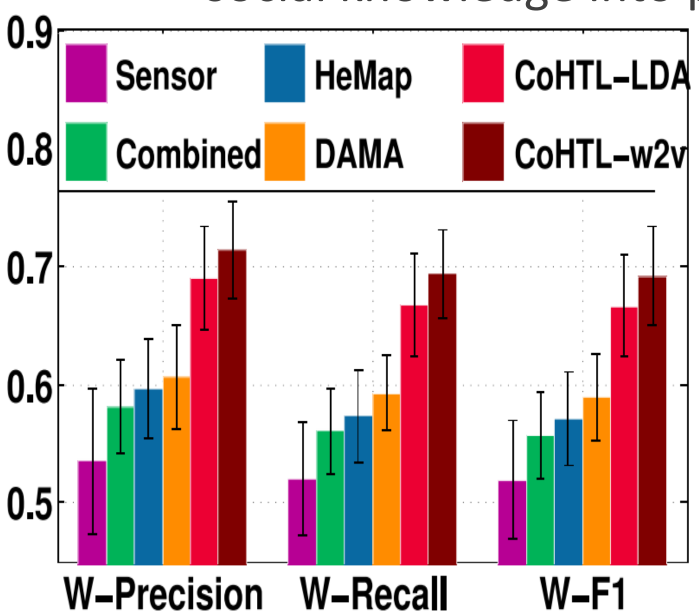
- Results

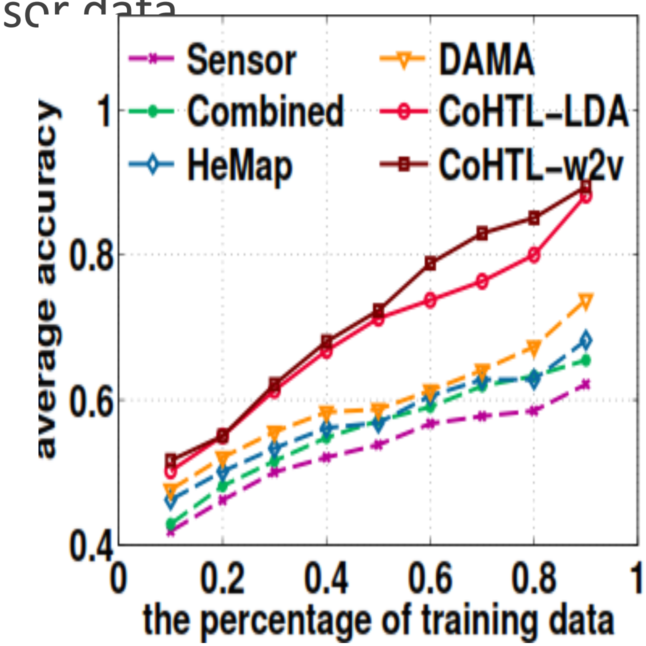
Heterogeneous transfer learning methods show

Our method could use only 50% labelled data of other methods to achieve the same performance.

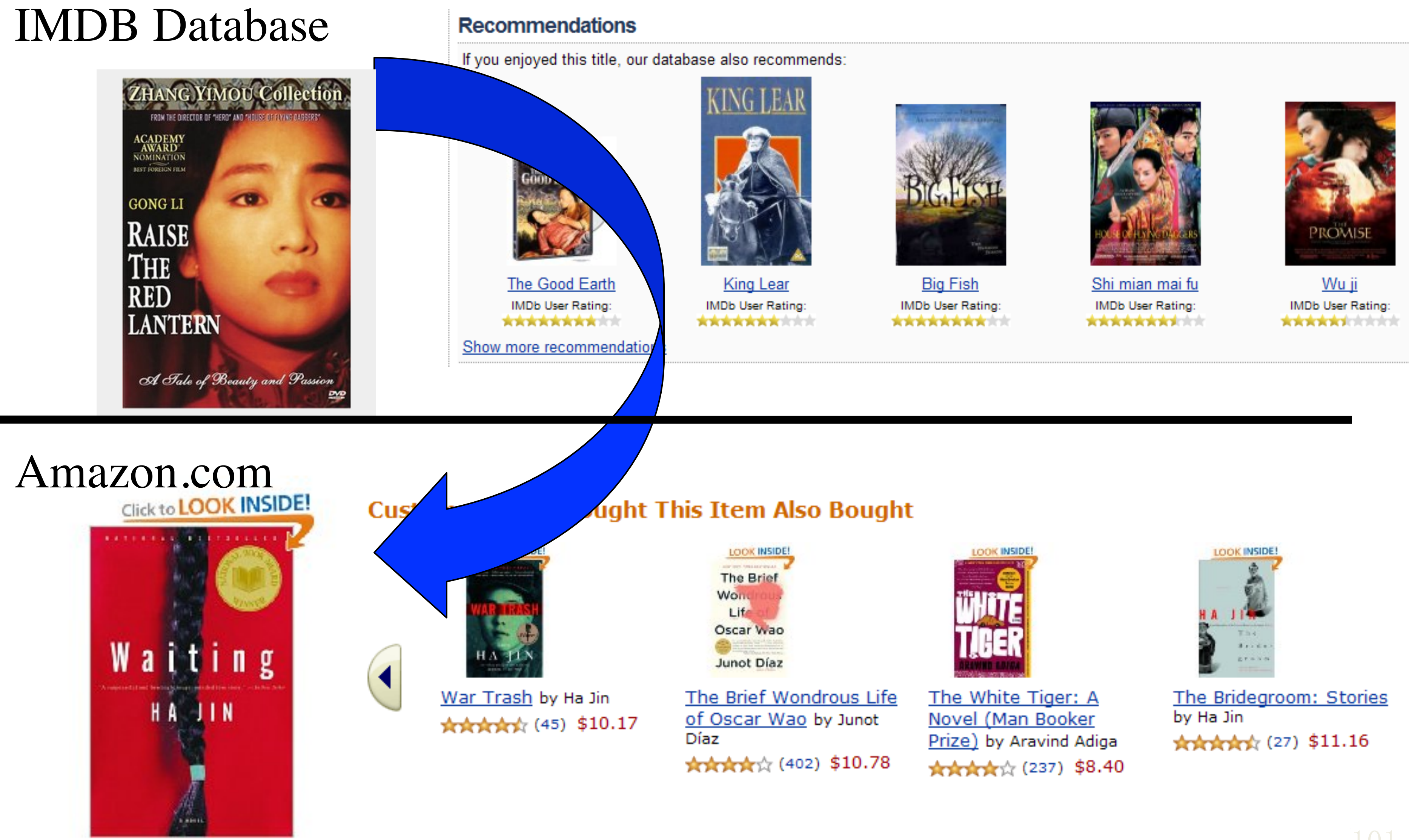
representation in a latent space is more effective than naive combination

v.s. Sensor), which validates the necessity of instilling social knowledge into physical sensor data.





Transfer Learning for Collaborative Filtering



Transfer Learning in Collaborative Filtering

- Source (Dense): Encode cluster-level rating patterns
- Target (Sparse): Map users/items to the encoded prototypes

MOVIES (Auxiliary - Dense)

	a	b	c	d	e	f
1	2	3	3	3	2	?
2	3	1	2	2	?	1
3	3	1	?	2	3	1
4	?	2	1	1	3	2
5	2	3	3	3	2	3
6	3	2	1	?	3	2

Permute rows & cols

	a	e	b	f	c	d
2	3	?	1	1	2	2
3	3	3	1	1	?	2
1	2	2	3	?	3	3
5	2	2	3	3	3	3
4	?	3	2	2	1	1
6	3	3	2	2	1	?

Reduce to Groups

	A	B	C
I	3	1	2
II	2	3	3
III	3	2	1

Cluster-level Rating Pattern Matching

BOOKS (Target - Sparse)

	a	b	c	d	e
1	?	1	3	3	1
2	3	3	2	?	3
3	2	2	?	3	?
4	1	1	3	?	?
5	1	?	?	3	1
6	3	?	2	2	3
7	?	2	3	3	2

	c	d	a	b	e
1	3	3	?	1	1
4	3	?	1	1	?
5	?	3	1	?	1
2	2	?	3	3	3
6	2	2	3	?	3
3	?	3	2	2	?
7	3	3	?	2	2

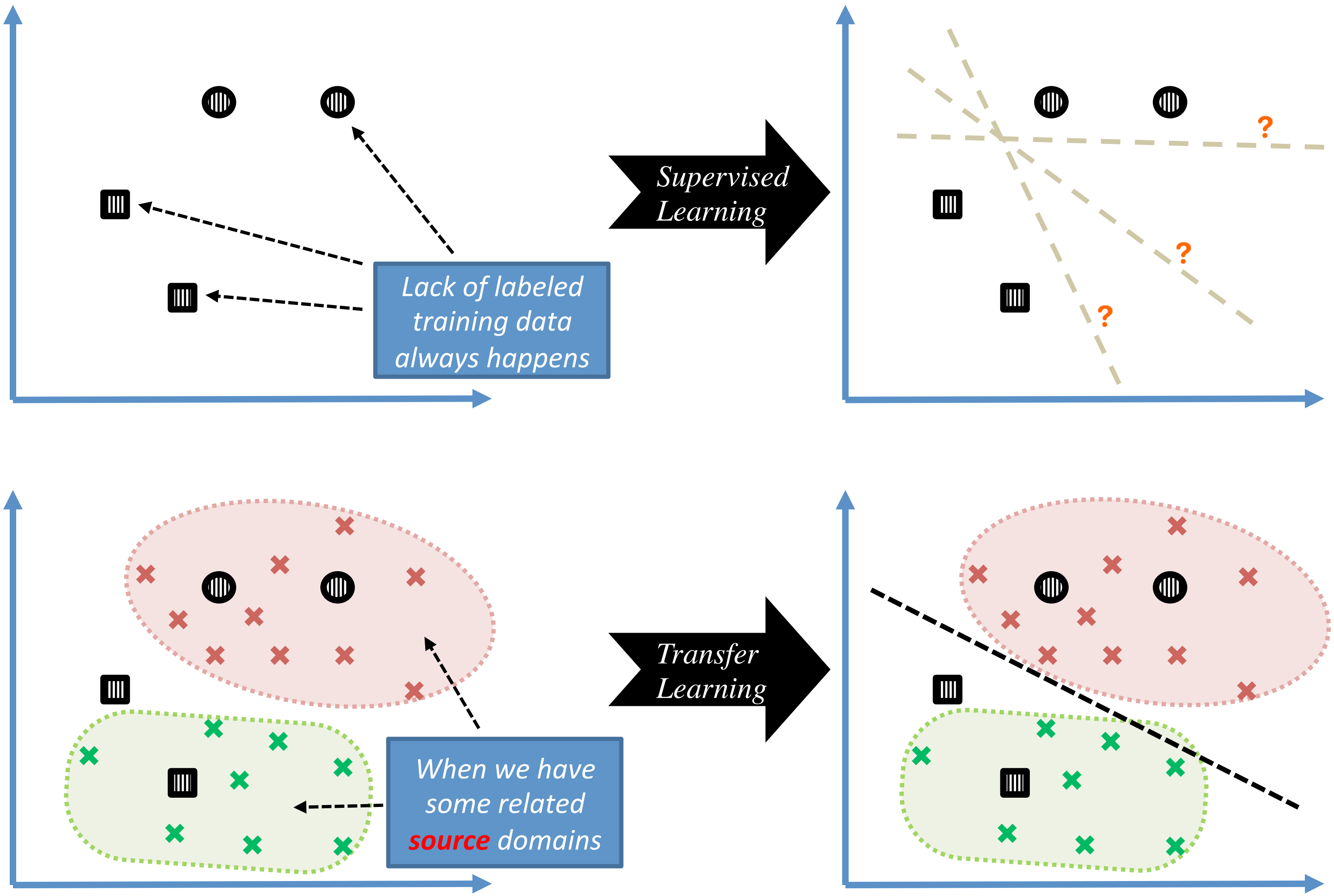
	A	B
I	3	1
II	2	3
III	3	2

ADVANCED DEVELOPMENTS

Source-Free Transfer Learning

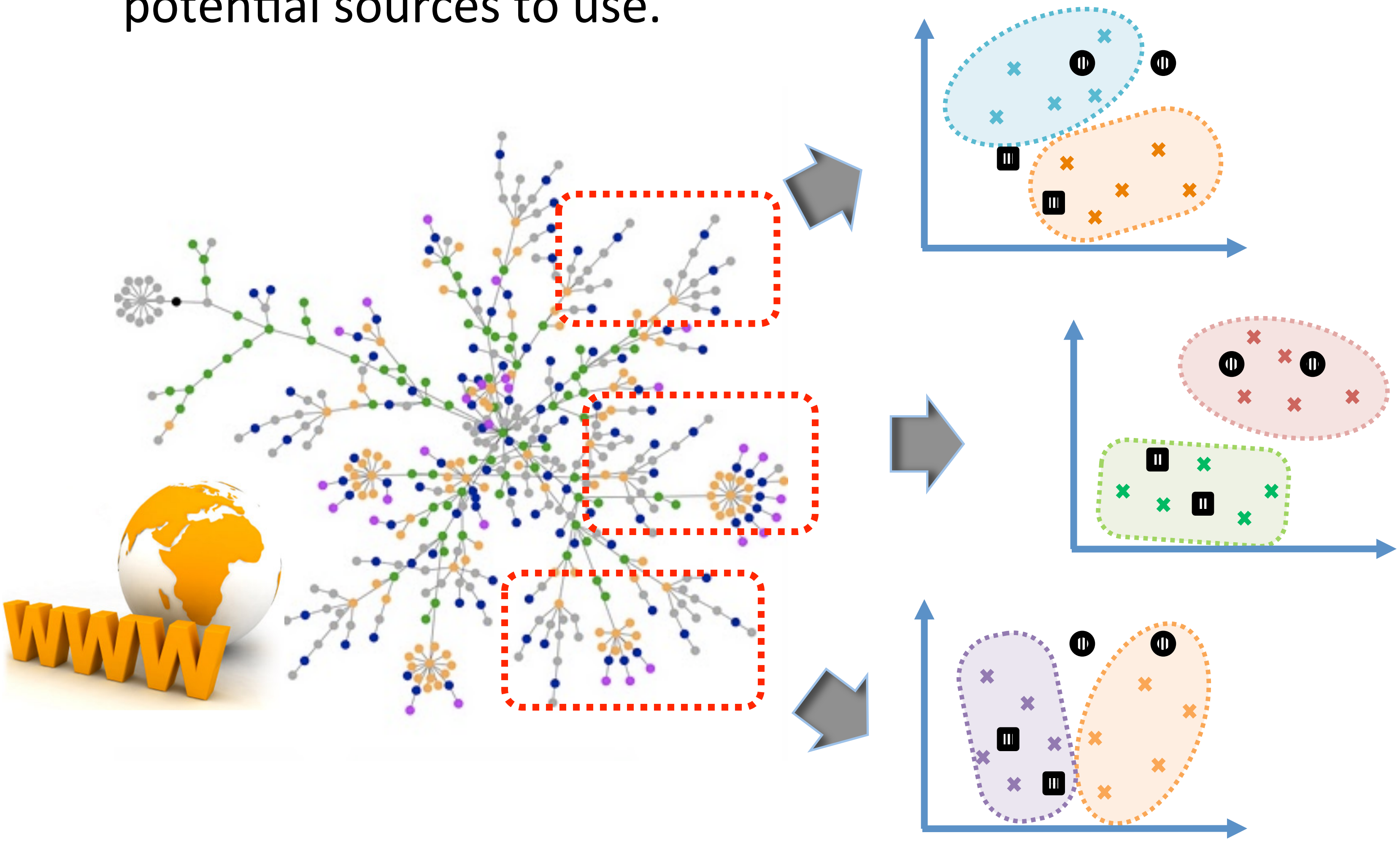
Evan Wei Xiang, Sinno Jialin Pan, Weike Pan, Jian Su and Qiang Yang. Source-Selection-Free Transfer Learning. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11), Barcelona, Spain, July 2011.

Transfer Learning

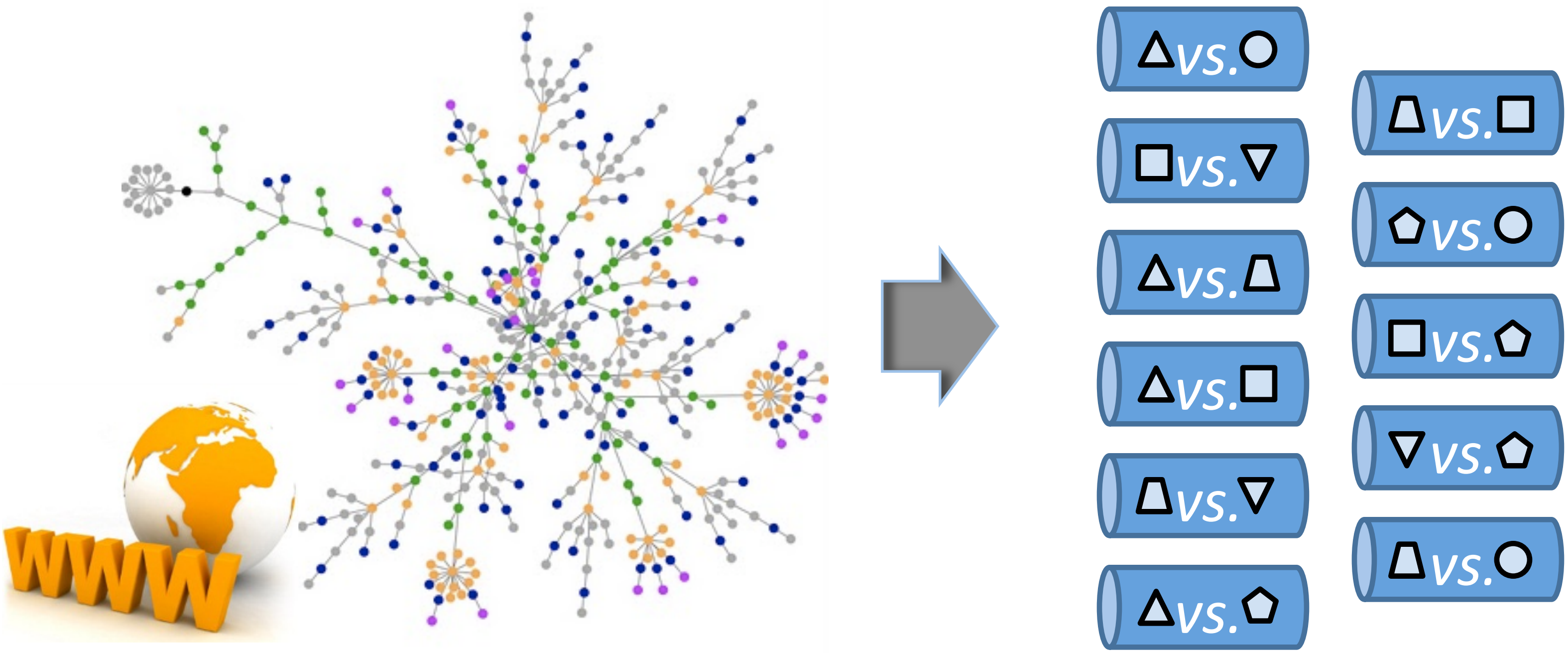


Where are the “right” source data?

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



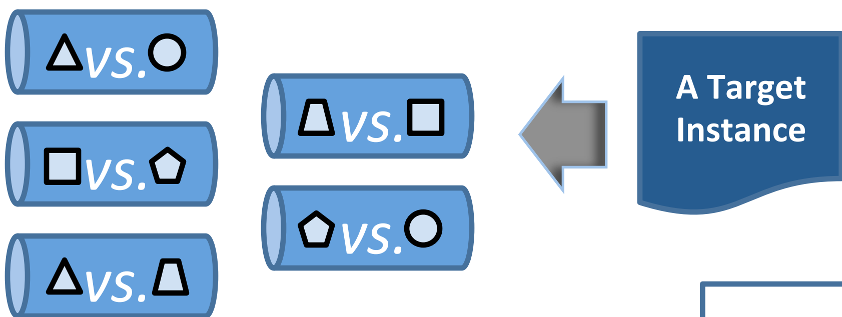
SFTL – Building base models



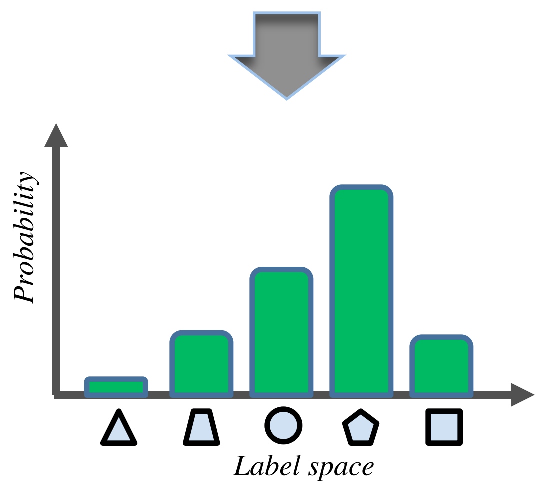
From the taxonomy of the online information source, we can “compile” a lot of base classification models

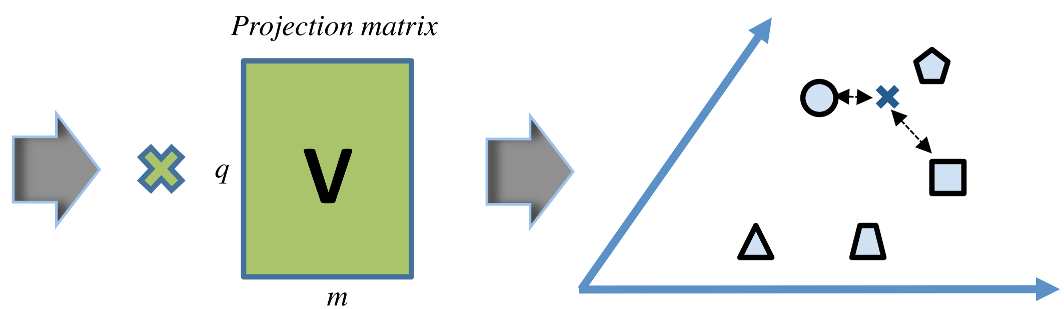
Source Free Transfer Learning

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



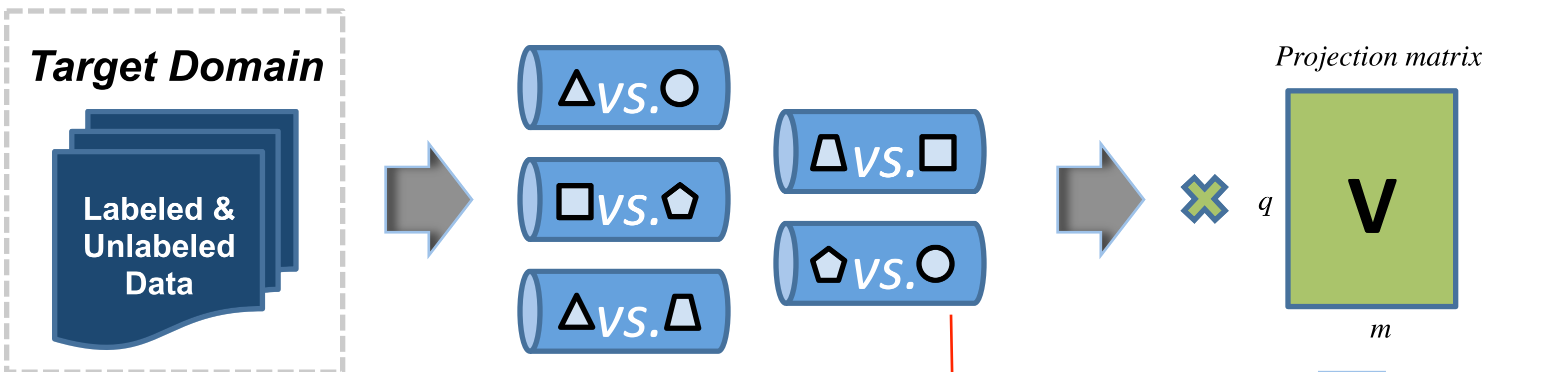
Then we can use the projection matrix V to transform such combined results from the label space to a latent space





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

Compilation: Learning a projection matrix ***W*** to amp the target instance to latent space



For each target instance, we first aggregate its prediction in 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

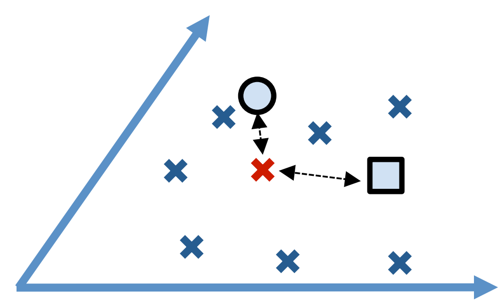
$$\Omega_{\mathbf{D}^u_T}(\mathbf{W}) = \frac{1}{n-\ell}\|\mathbf{W}'\mathbf{X}^u - \mathbf{V}'\mathbf{F}^u_S\|_F^2$$

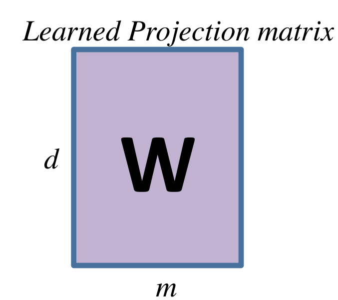
Loss on labeled data

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

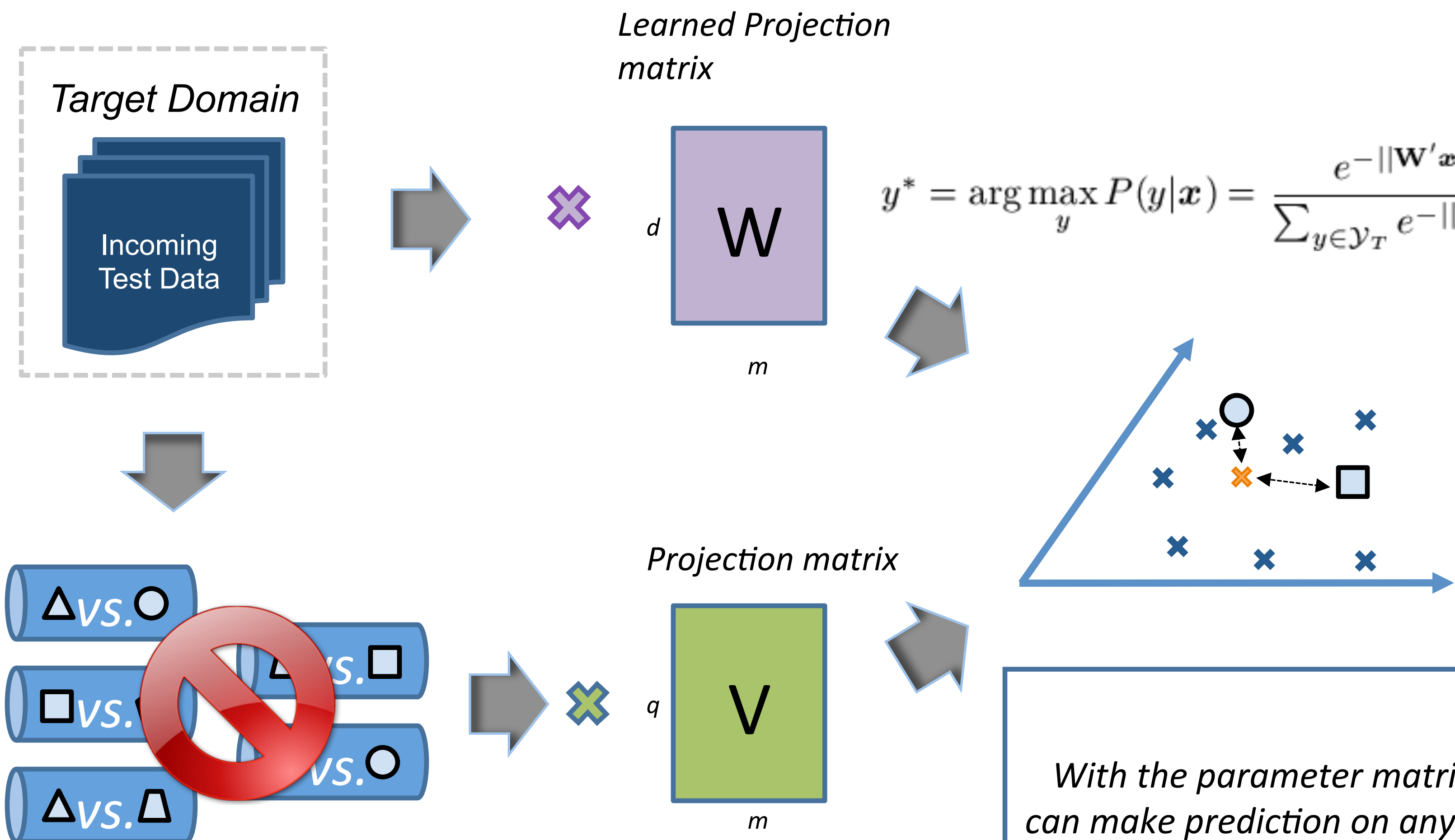
Our regression model

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





SFTL – Predictions for the incoming test data



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

No need to use base models explicitly!

With the parameter matrix W, we can make prediction on any incoming test data based on the distance to the label prototypes, without calling the base classification models

Transitive Transfer Learning

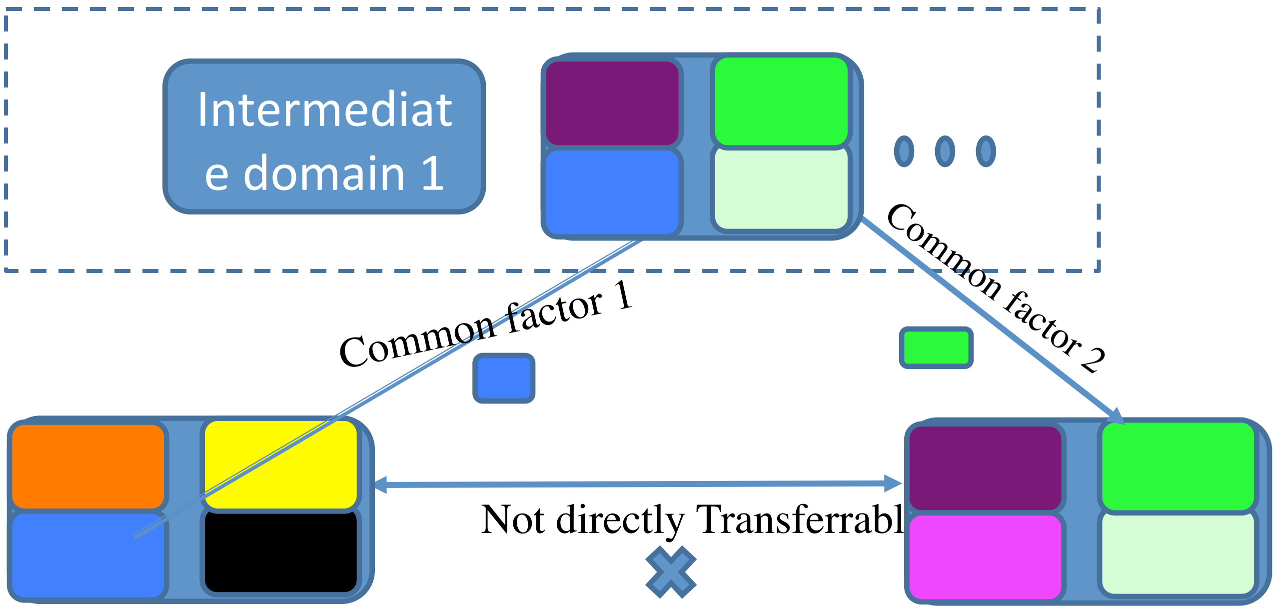
with intermediate domains

Qiang Yang
Hong Kong University of Science and Technology
http://www.cse.ust.hk/~qyang

Far Transfer vs. Near Transfer

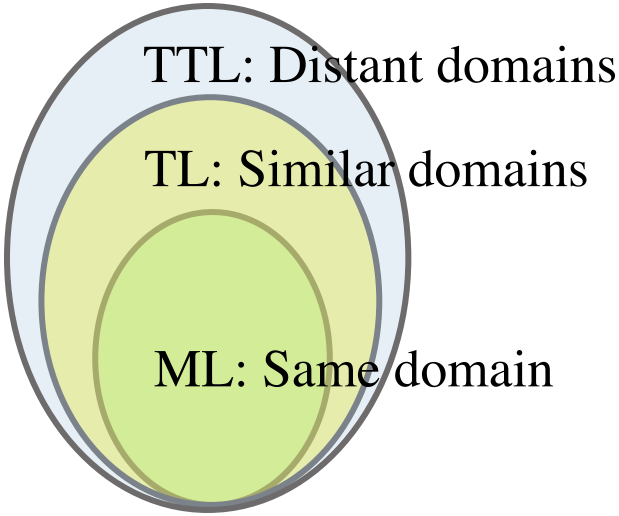
Problem definition

- Given **distant** source and target domains, and a set of intermediate domains, can we find one or more **intermediate domains** to enable the transfer learning between source and target?



Previous work and TTL

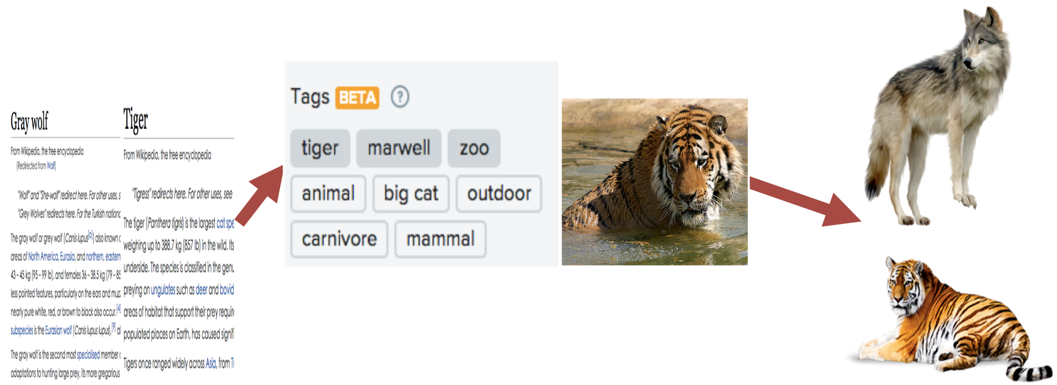
- Traditional machine learning
  - ✓ training and test data should be from the same problem domain.
- Transfer learning
  - ✓ training and test data should be from similar problem domains.
- Transitive transfer learning
  - ✓ training and test data could be from distant problem domains.



Text-to-Image Classification

Source and target domains have few overlaps

Text-to-image Classification with co-occurrence data as intermediate domain



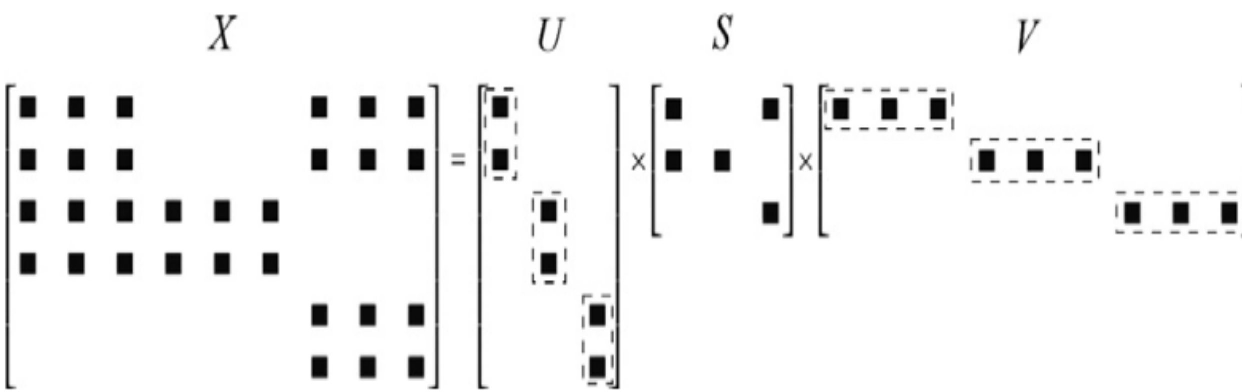
accelerator-to-gyroscope activity recognition with data from intelligent devices as intermediate domains

TTL: single intermediate domain

Intermediate domain selection, then propagate knowledge

- Crawl a lot of images with annotations from Internet
- Use domain distance, such as A-distance, to identify domain
- Transitive transfer through shared hidden factors in row by matrix tri-factorization

Matrix tri-factorization for clustering/classification



TTL: shared hidden factors in row by matrix tri-factorization

Coupled the two knowledge transfer processes

$$\min_{U,S,V\geq 0} \left\| X_s - \begin{bmatrix} U_{si} \\ U_{ss} \end{bmatrix}^T \begin{bmatrix} S_{si} \\ S_{ss} \end{bmatrix} V_s^T \right\|_F^2 + \left\| X_i - \begin{bmatrix} U_{si} \\ U_{ii} \end{bmatrix}^T \begin{bmatrix} S_{si} \\ S_{ii} \end{bmatrix} V_i^T \right\|_F^2 + \left\| X_i - \begin{bmatrix} U_{it} \\ U'_{ii} \end{bmatrix}^T \begin{bmatrix} S_{it} \\ S'_{ii} \end{bmatrix} V_i^T \right\|_F^2 + \left\| X_t - \begin{bmatrix} U_{it} \\ U'_{tt} \end{bmatrix}^T \begin{bmatrix} S_{it} \\ S'_{tt} \end{bmatrix} V_t^T \right\|_F^2$$

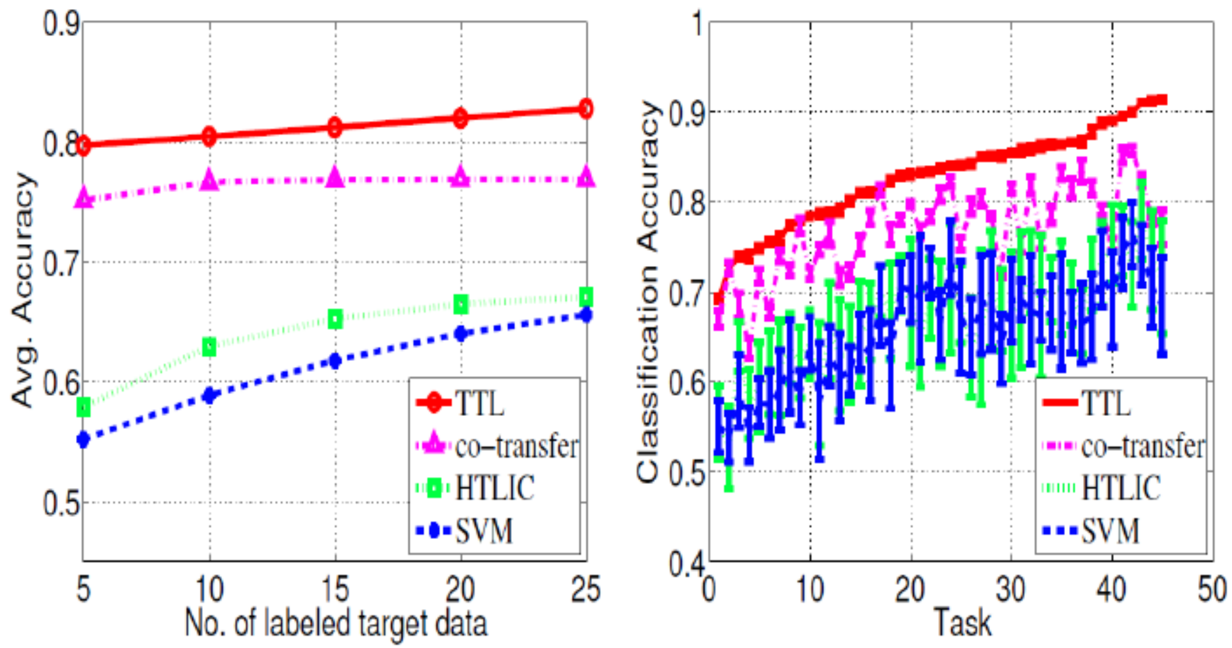
Transfer Knowledge between the source and intermediate domains

Transfer Knowledge between the intermediate and target domains

$$s.t.\, U_k^T \mathbf{1} = \mathbf{1}, \mathbf{V}_k^{\mathrm{T}} \mathbf{1} = \mathbf{1}, k \in \{s, i, t\}$$

Experiments NUS-WISE data set

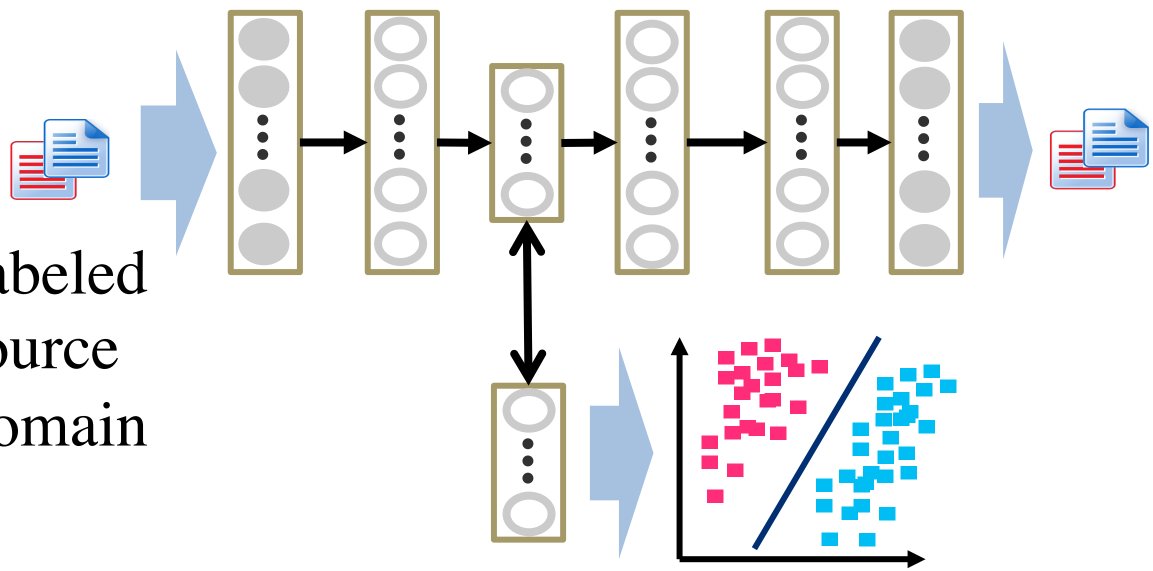
- The NUS-WISE data set are used
  - 45 text-to-image tasks
  - Each task is composed of 1200 text documents, 600 images, and 1600 co-occurred text-image pairs.



(a) Average performance (b) Detailed performance

Supervised Learning w/ auto-encoder

Feature Engineering

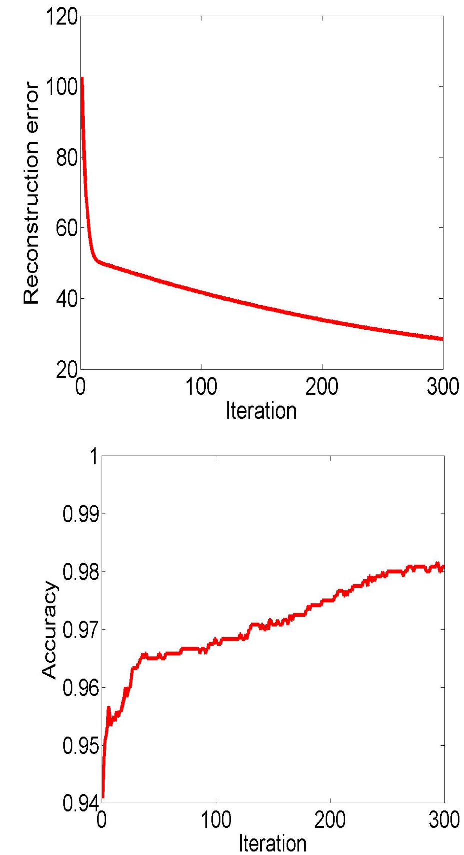


Labeled Source Domain

Shared

Predictive Model Learning

Text Classification



Designing Objective Function of TTL

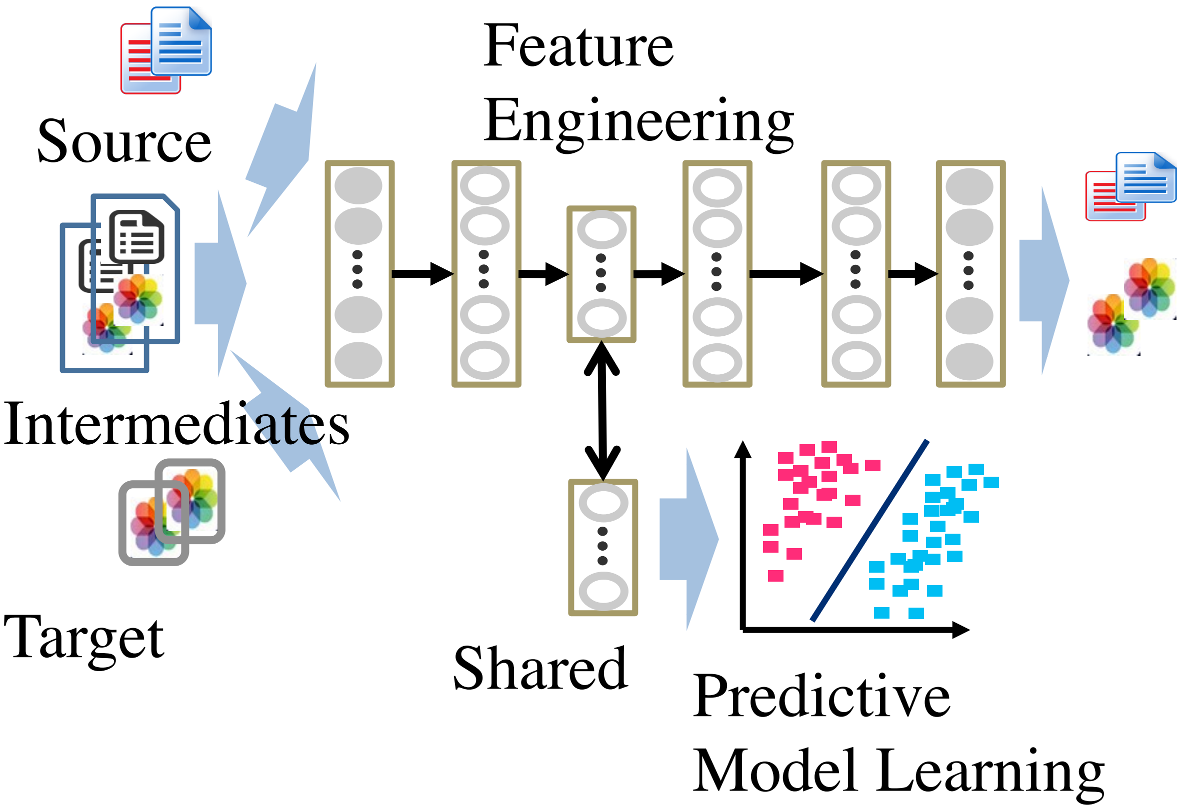
Transitive Transfer Learning with intermediate data

$$\underbrace{\sum_{i \in S} \left\| y_i - f_p\left(f_e\left(X_{0,j}, W^+\right), w_p\right) \right\|_2^2}_{\text{Predictive Model Learning}} + \underbrace{\sum_{j \in S,T} \sum_{j_i} \left\| f_r\left(X_{0,j_i}, W^+\right) - X_{c,j_i} \right\|_2^2}_{\text{Feature Engineering}} + \lambda \underbrace{\sum_{j \in D_1,\ldots,D_k} \beta_j \sum_{j_i} \left\| f_r\left(X_{0,j_i}, W^+\right) - X_{c,j_i} \right\|_2^2}_{\text{Intermediate domain weighting/selection}} + R(W^+, w_p, \beta_j)$$

The weights for the intermediate domains are learned from data.

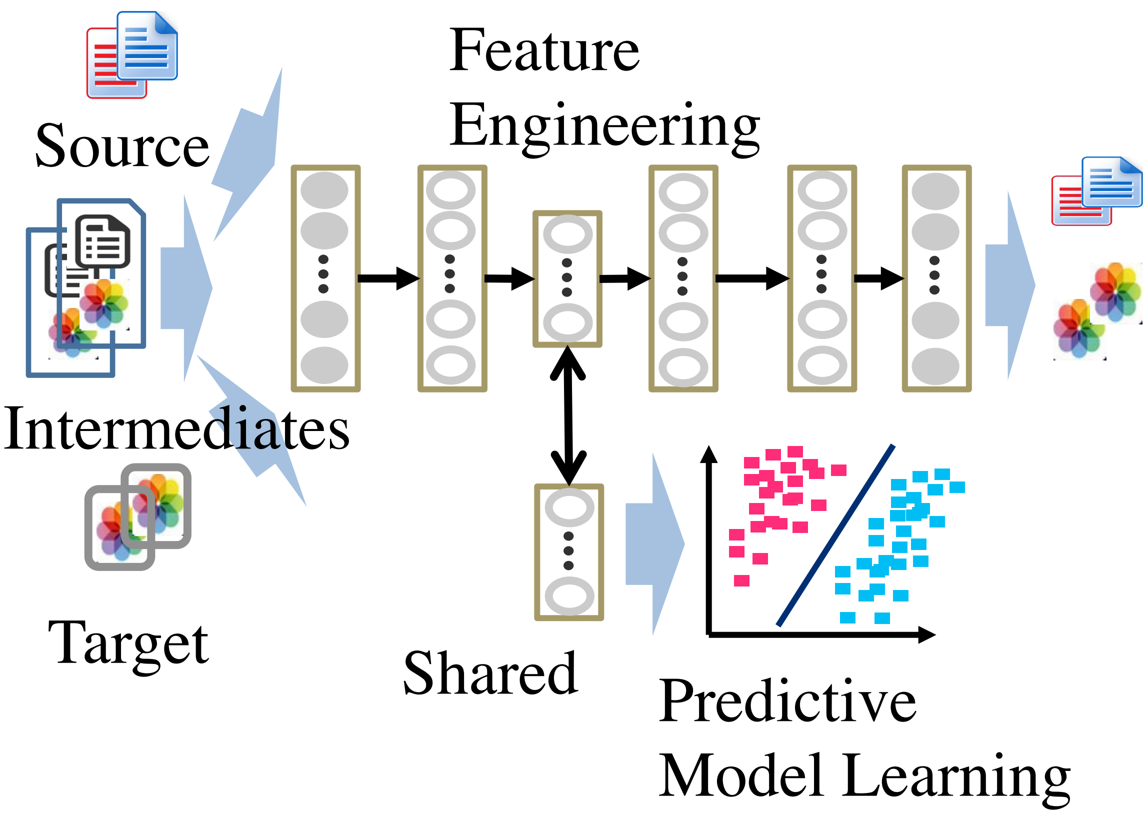
The intermediate data help find a better hidden layer.

TTL with supervised auto-encoder

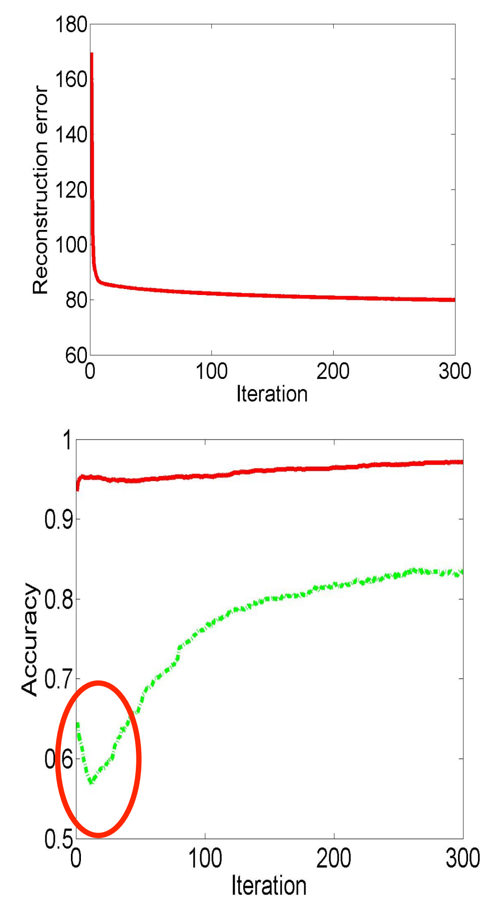


- The NUS-WISE data
  - 45 text-to-image tasks
  - Each task is composed of 1200 text documents, 600 images, and 1600 co-occurred text-image pairs. In each task, 1600*45 co-occurred text-image pairs will be used for knowledge transfer.

TTL with supervised auto-encoder



Text-to-image w/ intermediate data



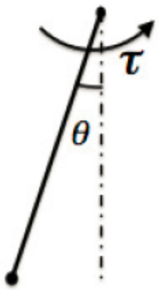
Reinforcement Transfer Learning via Sparse Coding

- Slow learning speed remains a fundamental problem for reinforcement learning in complex environments.
- Main problem: the numbers of states and actions in the source and target domains are different.
  - Existing works: hand-coded inter-task mapping between state-action pairs
- Tool: new transfer learning based on sparse coding

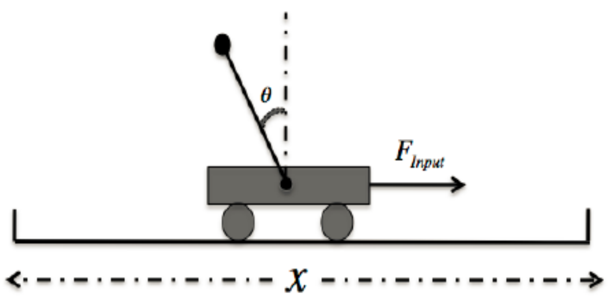
Ammar, Tuyls, Taylor, Driessens, Weiss: Reinforcement Learning Transfer via Sparse Coding. AAMAS, 2012.

Reinforcement Learning Transfer via Sparse Coding

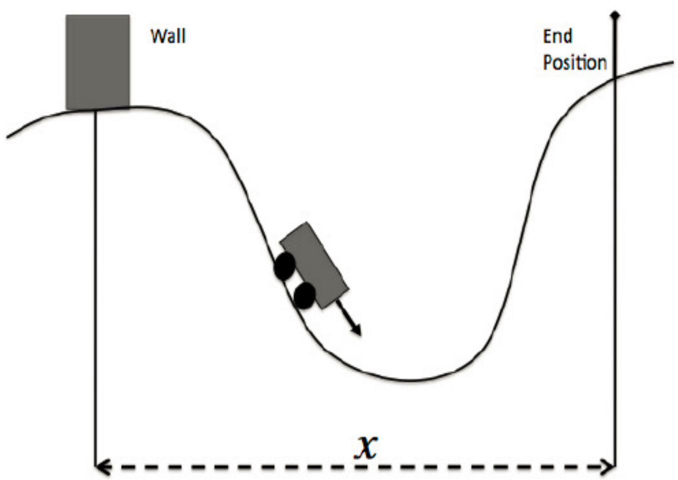
Authors measured the performance as the number of steps during an episode to control the pole in an upright position on a given fixed amount of samples.



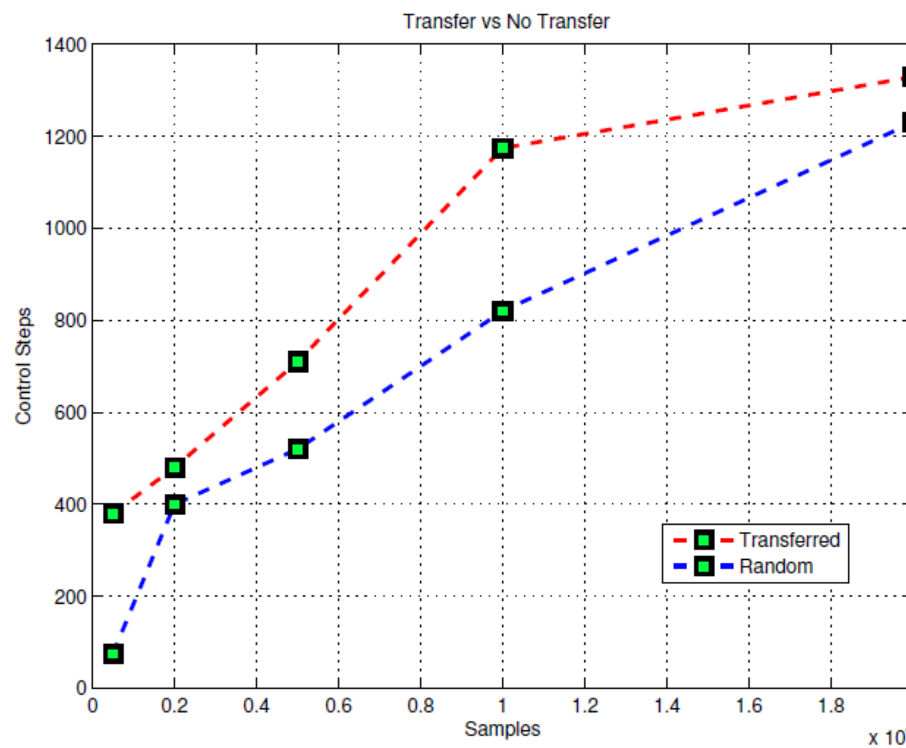
(a) Inverted Pendulum



(b) Cart Pole



(c) Mountain Car



Reinforcement Transfer Learning via Sparse Coding

- Given State-Action-State Triplets in the source task, learn dictionary as

$$\min_{\{\mathbf{b}_j\},\{a_j^{(i)}\}} \sum_{i=1}^{m} \frac{1}{2\sigma^2} ||\langle s_0, a_0, s_0^{'} \rangle^{(i)} - \sum_{j=1}^{d_1} \mathbf{b}_j a_j^{(i)}||_2^2 + \beta \sum_{i=1}^{m} \sum_{j=1}^{d_1} ||a_j^{(i)}||_1 \;\; s.t. \;\; ||\mathbf{b}_j||_2^2 \leq c, \forall j = \{1, 2, \ldots, d_1\}$$

- Using the coefficient matrix in the first step, we can learn the dictionary in the target task as

$$\min_{\{\mathbf{z}_j\},\{c_j^{(i)}\}} \sum_{i=1}^{m} \frac{1}{2\sigma^2} ||\langle \mathbf{a}_{1:d_1} \rangle^{(i)} - \sum_{j=1}^{d_n} \mathbf{z}_j c_j^{(i)}||_2^2 + \beta \sum_{i=1}^{m} \sum_{j=1}^{d_n} ||c_j^{(i)}||_1 \;\; s.t. \;\; ||\mathbf{z}_j||_2^2 \leq o, \forall j = \{1, 2, \ldots, d_n\}$$

- Then for each triplet in the target task, - sparse projection is used to find its coefficients

$$\hat{\phi}^{(i)}(\langle s_t, a_t, s_t^{'} \rangle) = \arg\min_{\phi^{(i)}} ||\langle s_t, a_t, s_t^{'} \rangle^{(i)} - \sum_{j=1}^{d_n} \phi_j^{(i)} \mathbf{z}_j||_2^2 + \beta ||\phi^{(i)}||_1$$

- As a result, the inter-task mapping can be learned!

Reference

- [Thorndike and Woodworth, The Influence of Improvement in one mental function upon the efficiency of the other functions, 1901]
- [Taylor and Stone, Transfer Learning for Reinforcement Learning Domains: A Survey, JMLR 2009]
- [Pan and Yang, A Survey on Transfer Learning, IEEE TKDE 2009]
- [Quionero-Candela, *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]
- [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]

Reference (cont.)

- [Raina *etal*., Self-taught Learning: Transfer Learning from Unlabeled Data, ICML 2007]
- [Dai *etal.,* Boosting for Transfer Learning, ICML 2007]
- [Glorot *etal*., Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach, ICML 2011]
- [Davis and Domingos, Deep Transfer vis Second-order Markov Logic, ICML 2009]
- [Mihalkova *etal*., Mapping and Revising Markov Logic Networks for Transfer Learning, AAAI 2007]
- [Li *etal*., Cross-Domain Co-Extraction of Sentiment and Topic Lexicons, ACL 2012]

Reference (cont.)

- [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]
- [Cristianini *etal*., On Kernel Target Alignment, NIPS 2002]
- [Huang *etal*., Correcting Sample Selection Bias by Unlabeled Data, NIPS 2006]
- [Zadrozny, Learning and Evaluating Classifiers under Sample Selection Bias, ICML 2004]

Transfer Learning in Convolutional Neural Networks

- Convolutional neural networks (CNN): outstanding image-classification.
- Learning CNNs requires a very large number of annotated image samples
  - Millions of parameters, to many that prevents application of CNNs to problems with limited training data.
- Key Idea:
  - the internal layers of the CNN can act as a generic extractor of mid-level image representation
  - Model-based Transfer Learning

Thank You