AI6102: Machine Learning Methodologies & Applications

L13: Transfer Learning

Sinno Jialin Pan

Nanyang Technological University, Singapore

Homepage: http://www.ntu.edu.sg/home/sinnopan

NANYANG TECHNOLOGICAL UNIVERSITY

Outline

- What is transfer learning?
- Transfer learning methodologies
  - Instance-based
  - Feature-based
  - Parameter-based
  - Relational

Transfer of Learning

Psychological point of view

- Inspired by human's transfer of learning ability
- 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.

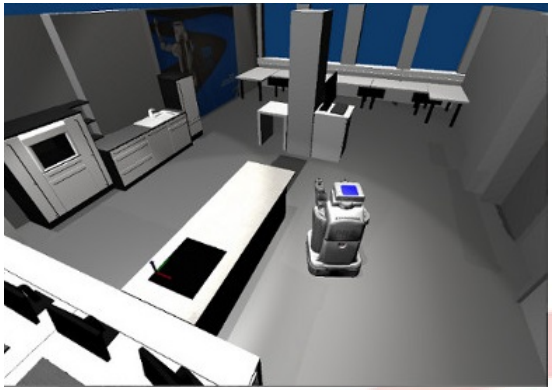
Transfer Learning

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

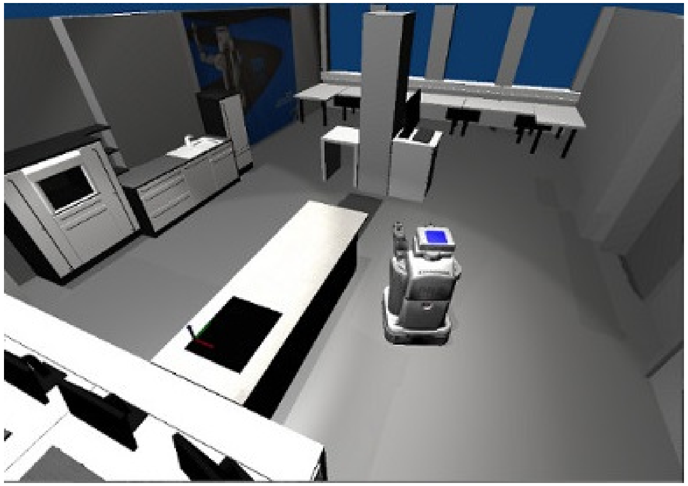
Motivating Example I

- Goal: to train a robot to accomplish Task $\boldsymbol{T}_1$ in an indoor environment $\boldsymbol{E}_1$ using machine learning techniques:
  - Sufficient training data required: sensor readings to measure the environment as well as supervision, i.e. labels by human or feedback from environment
  - A policy or predictive model can be learned, and used in the same environment



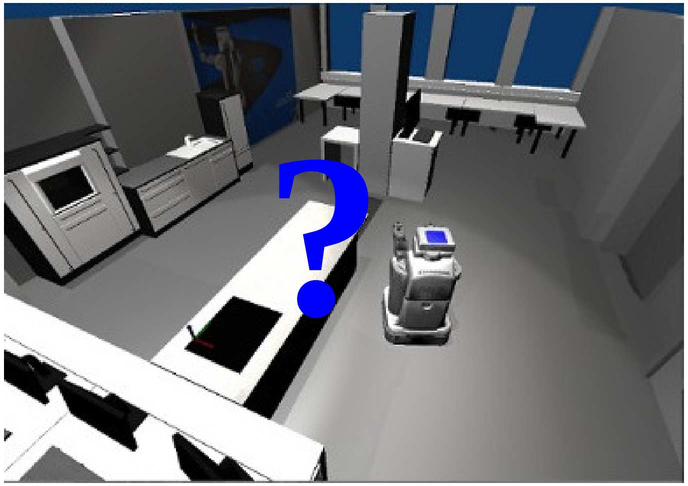
Task $\boldsymbol{T}_1$ in environment $\boldsymbol{E}_1$

To train the robot from scratch?
Expensive & time consuming!

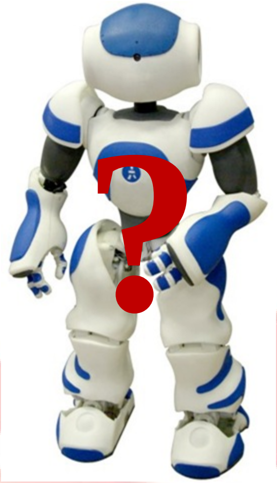


Task T_1

Environment changes E_2

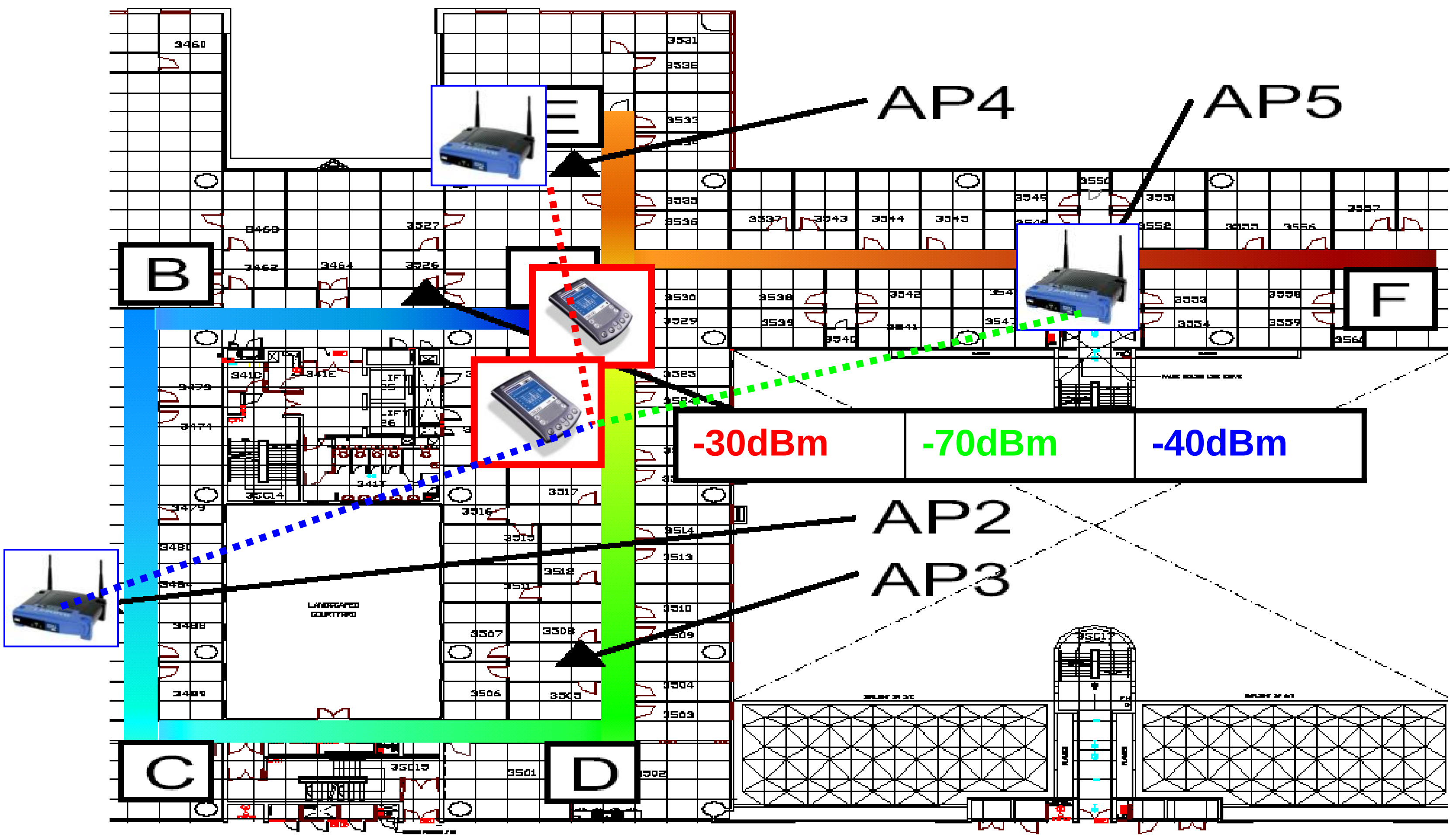


Task T_2



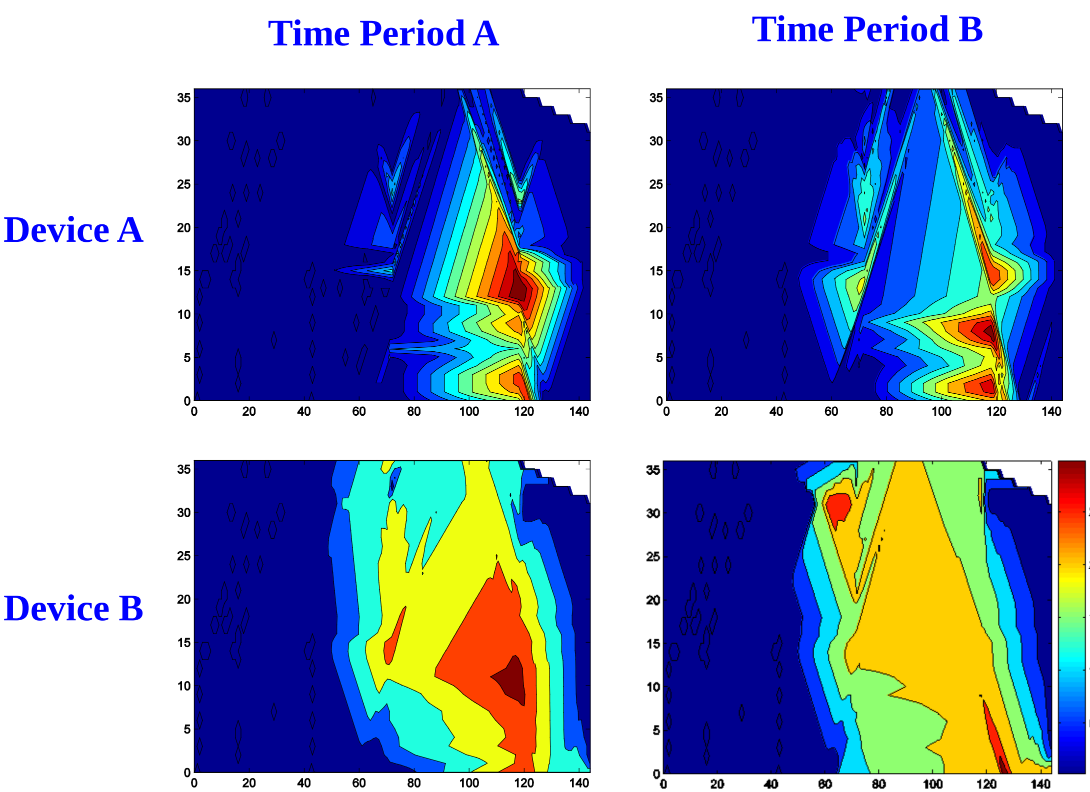
New robot

Motivating Example II

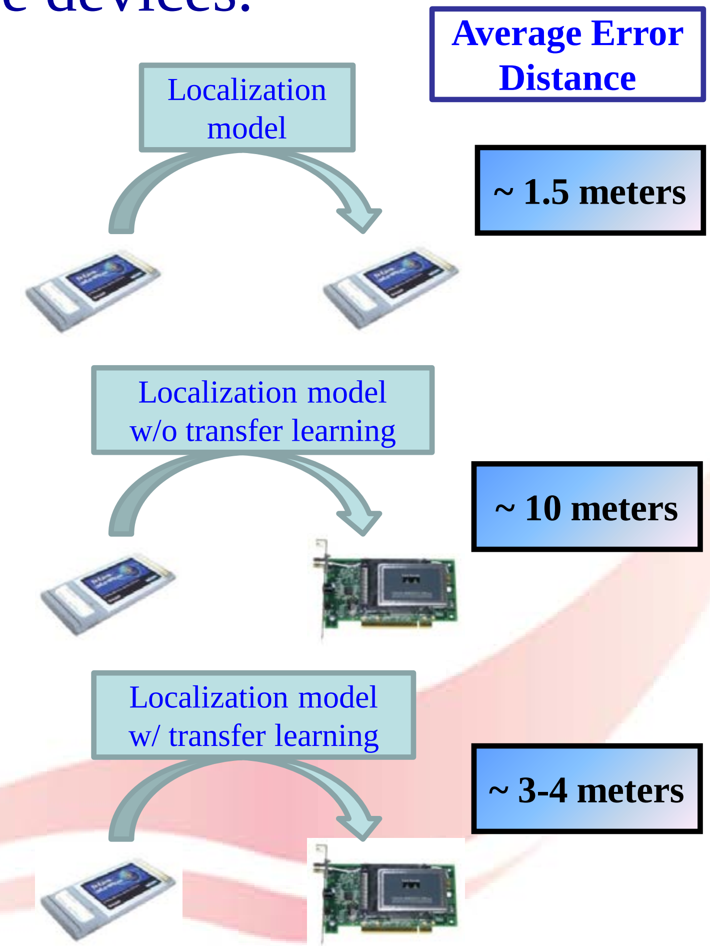


Motivating Example II (cont.)

- **WiFi localization:** signal strength changes a lot over different time periods, or across different mobile devices.



Contour of signal strength values

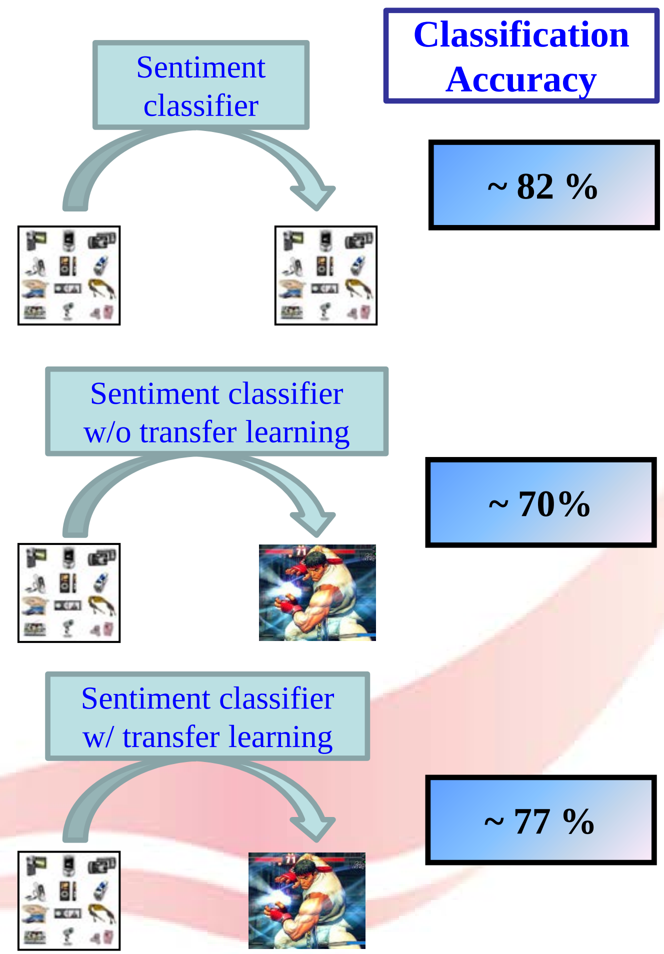


Motivating Examples III

- **Sentiment analysis:** users may use different sentiment words across different 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.

Product reviews on different domains



A Strong Assumption

- **Assumption:** training and test data are assumed to be
  - Represented in the same feature space, AND
  - Follow the same data distribution
- If the assumption holds, then there is theoretical guarantee on the performance of the model learned with training data on the test data

A Strong Assumption (cont.)

- Consider the learning framework of empirical risk minimization

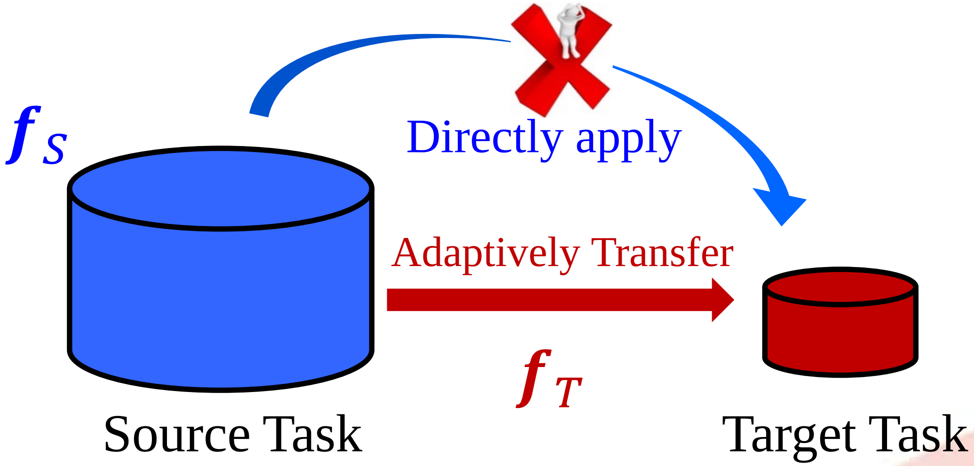
$$
\begin{aligned}
\theta^* &= \arg\min_{\theta} \mathbb{E}_{(x,y)\sim P_{tst}}[\ell(x,y;\theta)] \\
&= \arg\min_{\theta} \mathbb{E}_{(x,y)\sim P_{tst}}\left[\frac{P_{trn}(x,y)}{P_{trn}(x,y)}\ell(x,y;\theta)\right] \\
&= \arg\min_{\theta} \int_y \int_x P_{tst}(x,y)\left(\frac{P_{trn}(x,y)}{P_{trn}(x,y)}\ell(x,y;\theta)\right) dx\, dy \\
&= \arg\min_{\theta} \int_y \int_x P_{trn}(x,y)\left(\frac{P_{tst}(x,y)}{P_{trn}(x,y)}\ell(x,y;\theta)\right) dx\, dy \\
&= \arg\min_{\theta} \mathbb{E}_{(x,y)\sim P_{trn}}\left[\boxed{\frac{P_{tst}(x,y)}{P_{trn}(x,y)}}^{\,=1}\ell(x,y;\theta)\right]
\end{aligned}
$$

If $P_{tst}(x,y) = P_{trn}(x,y)$

$$
= \arg\min_{\theta} \mathbb{E}_{(x,y)\sim P_{trn}}[\ell(x,y;\theta)]
$$

Transfer Learning

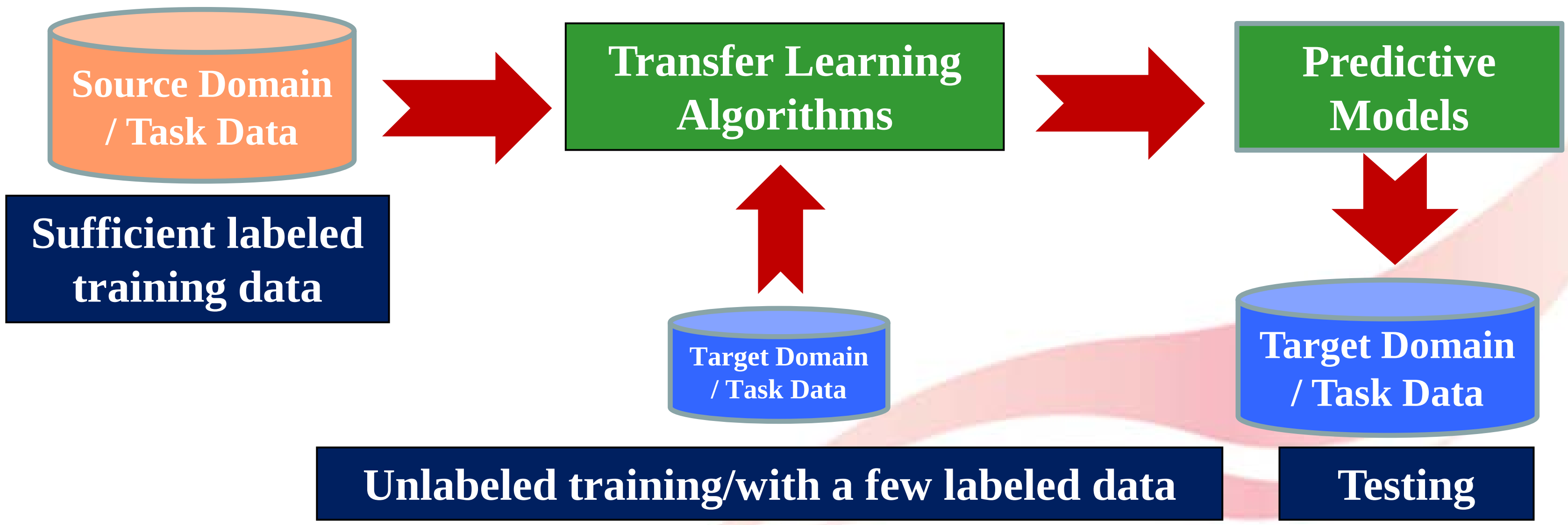
- **<u>In practice:</u>** training and test data come from different domains
  - Represented in different feature spaces, OR
  - Follow different data distributions
- $\arg\min_{\theta} \mathbb{E}_{(x,y)\sim P_{tst}}[\ell(x, y; \theta)] \neq \arg\min_{\theta} \mathbb{E}_{(x,y)\sim P_{trn}}[\ell(x, y; \theta)]$



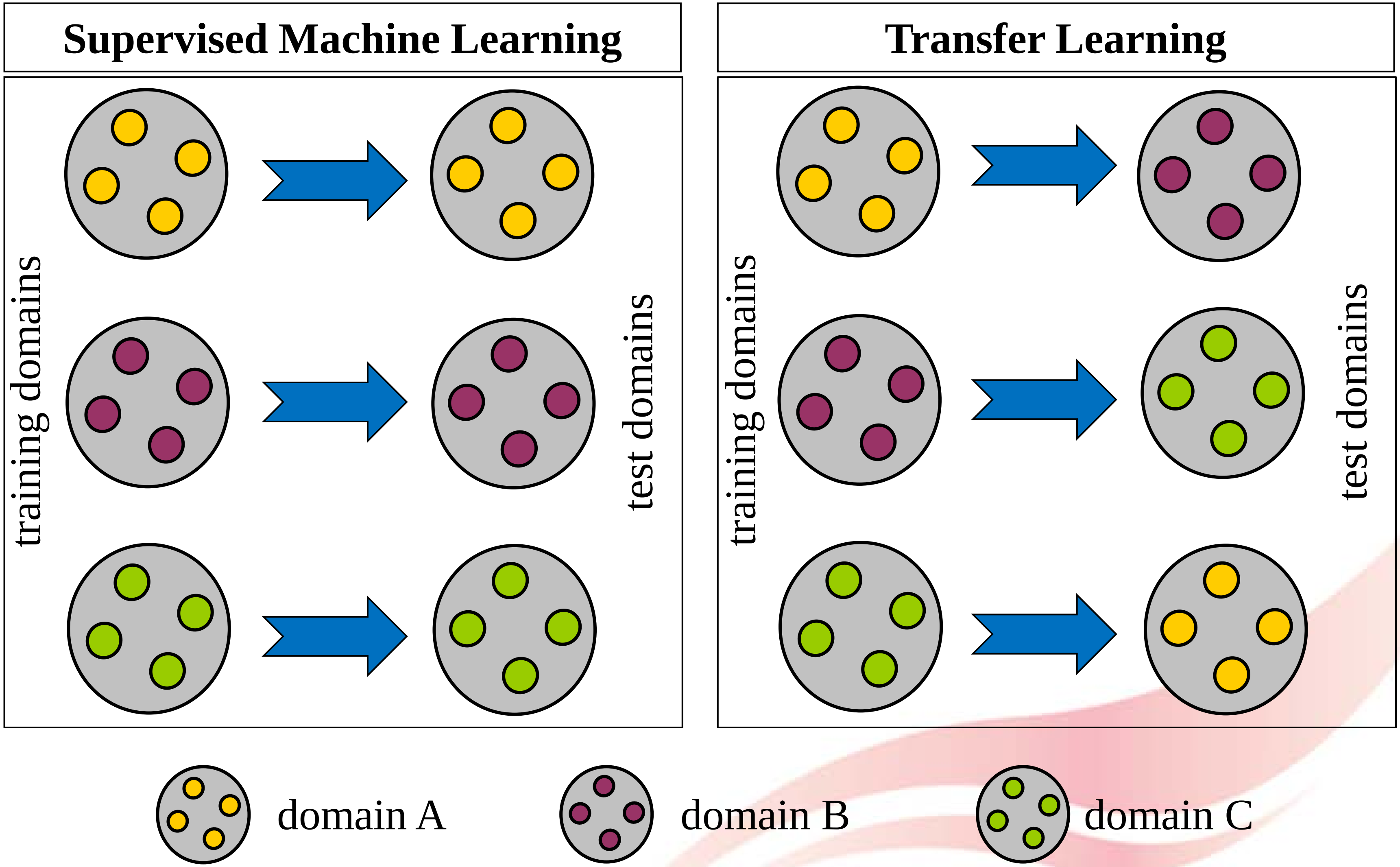
What if machines have transfer learning ability?

Transfer Learning (cont.)

- Given a target domain/task, transfer learning aims to
  1) identify the commonality between the target domain/task and previous domains/tasks
  2) transfer knowledge from the previous domains/tasks to the target one such that human supervision on the target domain/task can be dramatically reduced.

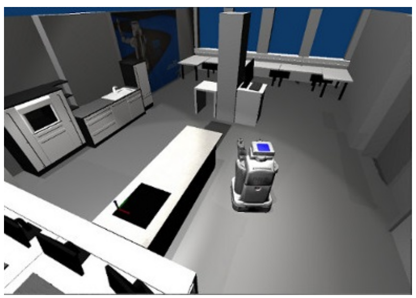


TL v.s. Supervised ML



TL for Different ML Problems

- Transfer learning for reinforcement learning



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

- Transfer learning for classification/regression

[**<u>Pan</u>** and Yang, A Survey on Transfer Learning, IEEE TKDE 2010]

[**<u>Pan</u>**, Transfer Learning, Chapter 21, Data Classification: Algorithms and Applications 2014]

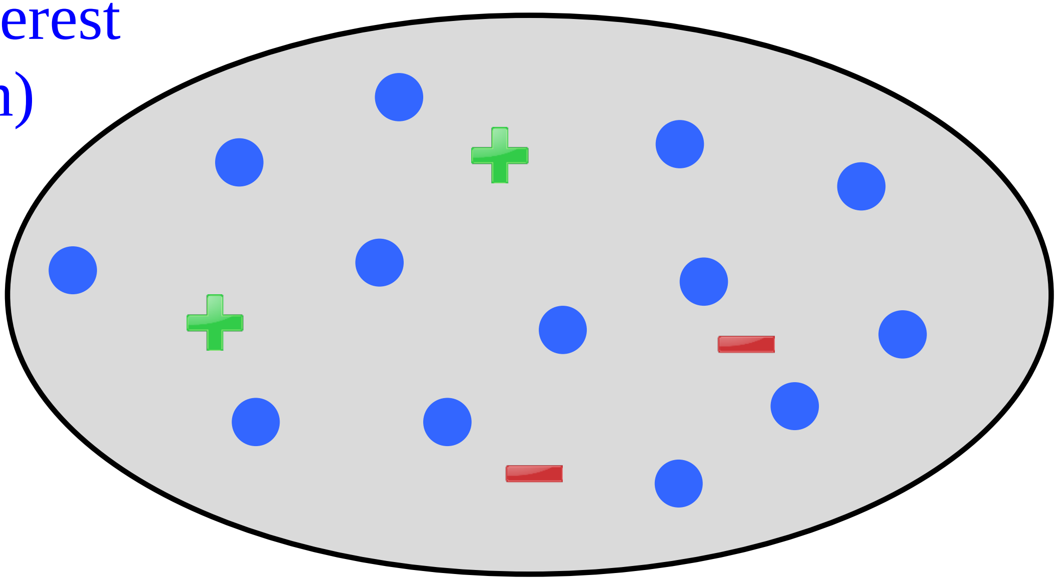
[Yang, Zhang, Dai and **<u>Pan</u>**, Transfer Learning, Cambridge University Press 2020]

TL v.s. Active Learning & Semi-supervised Learning

- They are all proposed to address the labeled data sparsity issue on the learning domain of interest
- The strategies used or the assumptions made are quite different
- They can be combined to further boost the performance of the learning problems with sparse labeled data

Semi-supervised Learning

Domain of interest
(target domain)

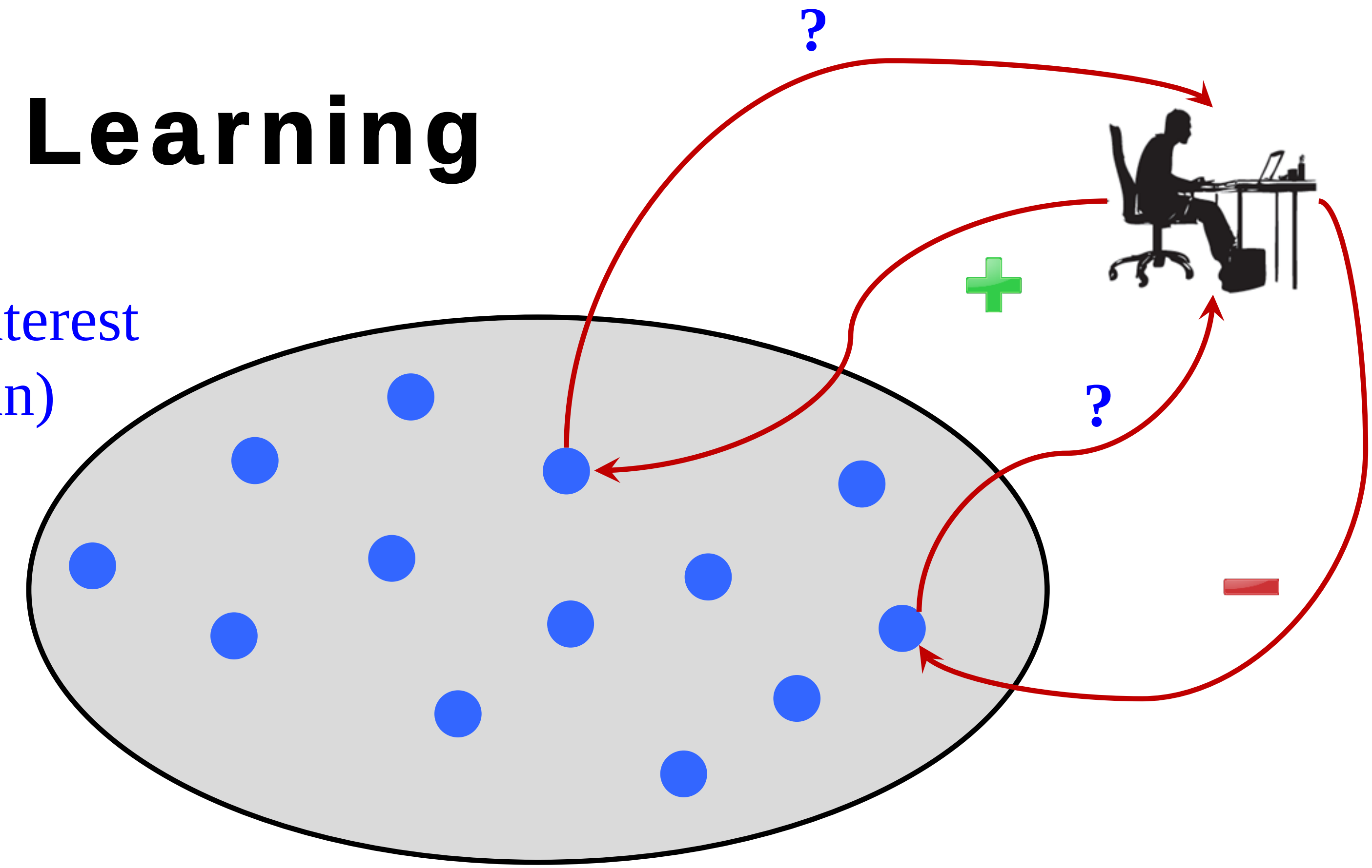


Assumption:

1. A little labeled data is available
2. Plenty of unlabeled data is cheap to collect
3. Underlying cluster or manifold structure can be discovered by using unlabeled data, and is useful for label propagation

Active Learning

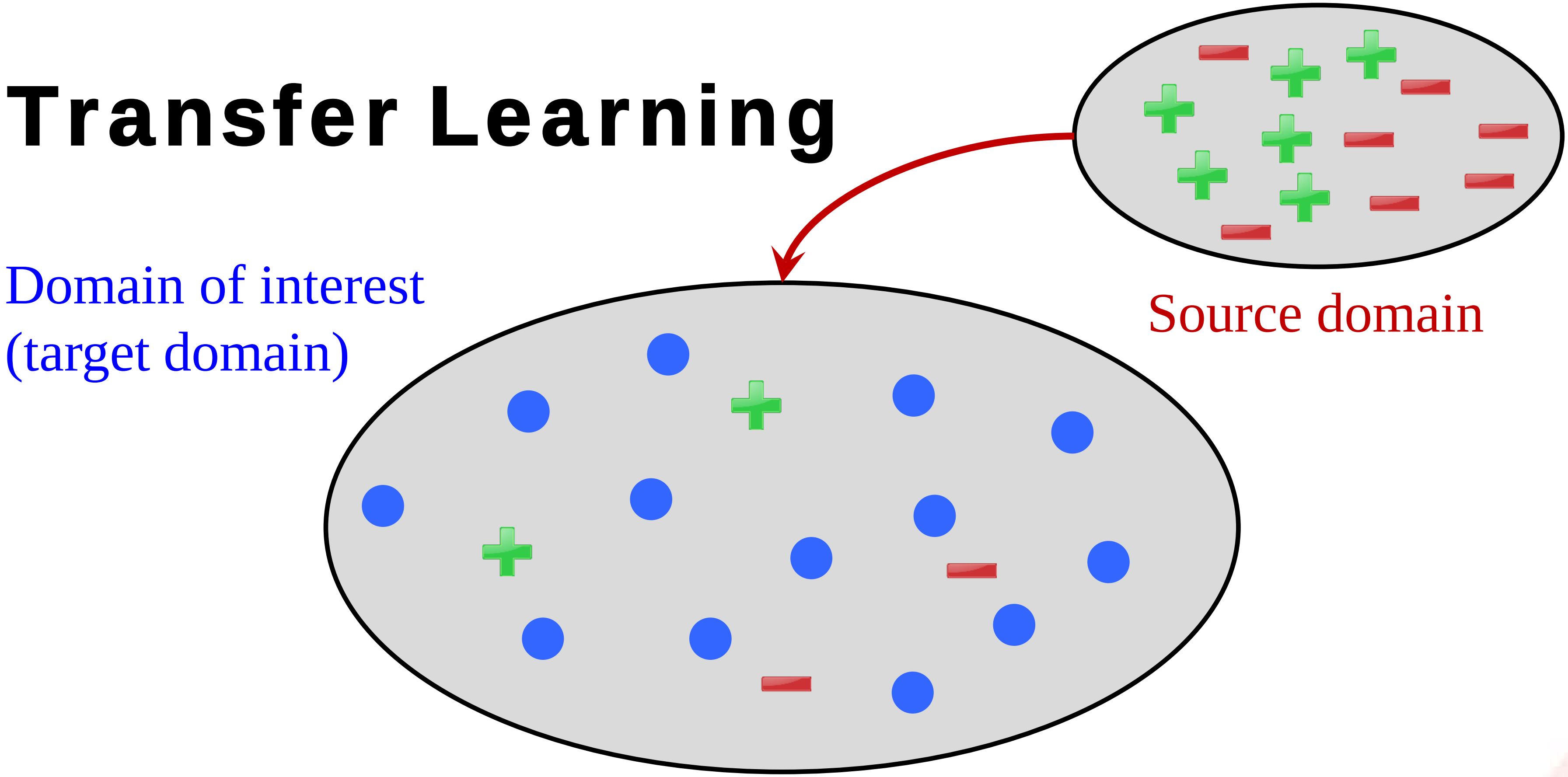
Domain of interest
(target domain)



Assumption:

1. A pool of unlabeled data is available
2. An oracle is able to provide labels via querying with cost
3. The budget for querying labels is limited

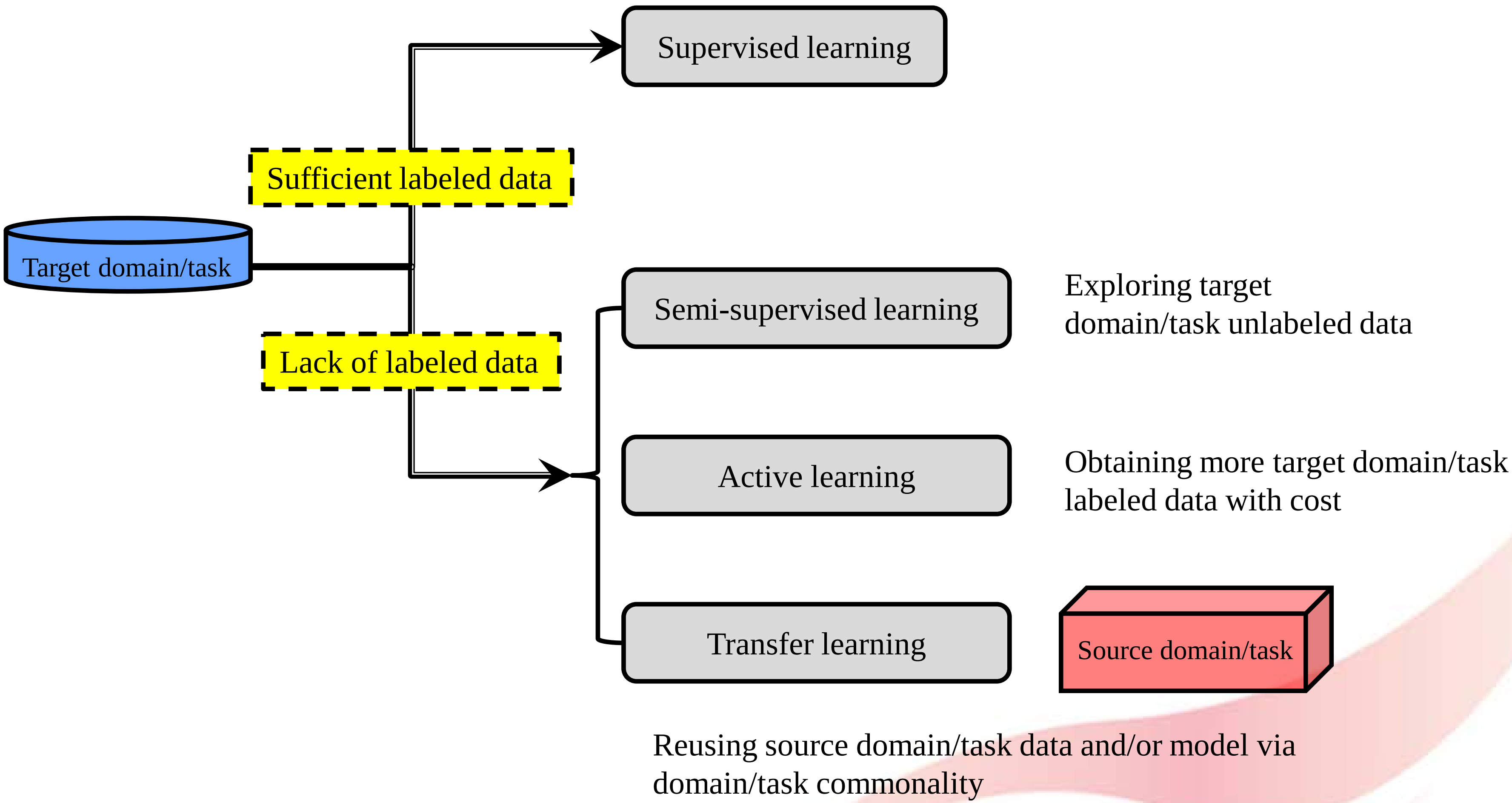
Transfer Learning



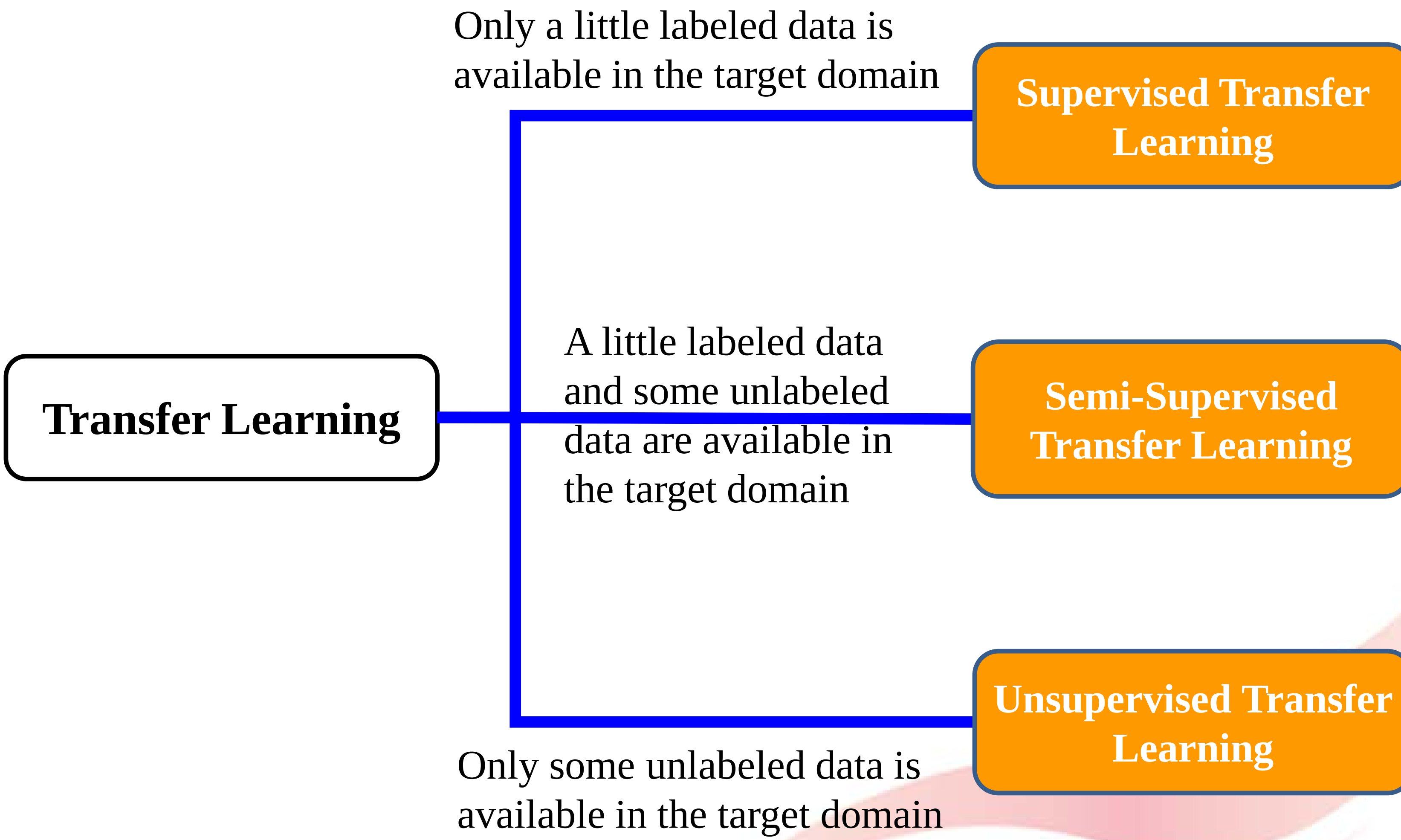
Assumption:

1. A little labeled or/and some unlabeled data is available on the target domain
2. Plenty labeled data is available on related source domain(s)
3. Source-domain data can be borrowed to learn a target classifier after some adaptation

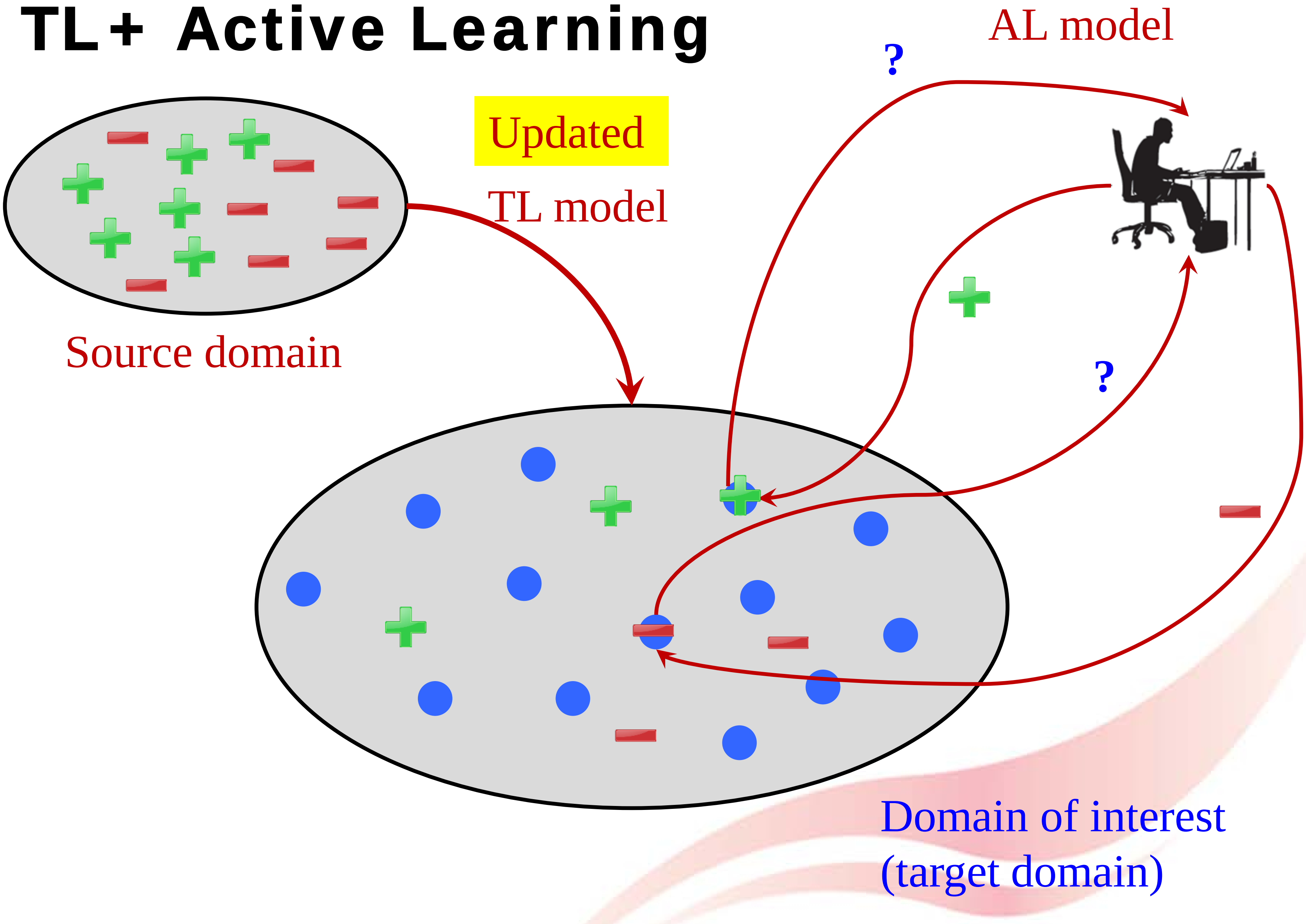
TL v.s. Active Learning & Semi-supervised Learning (cont.)



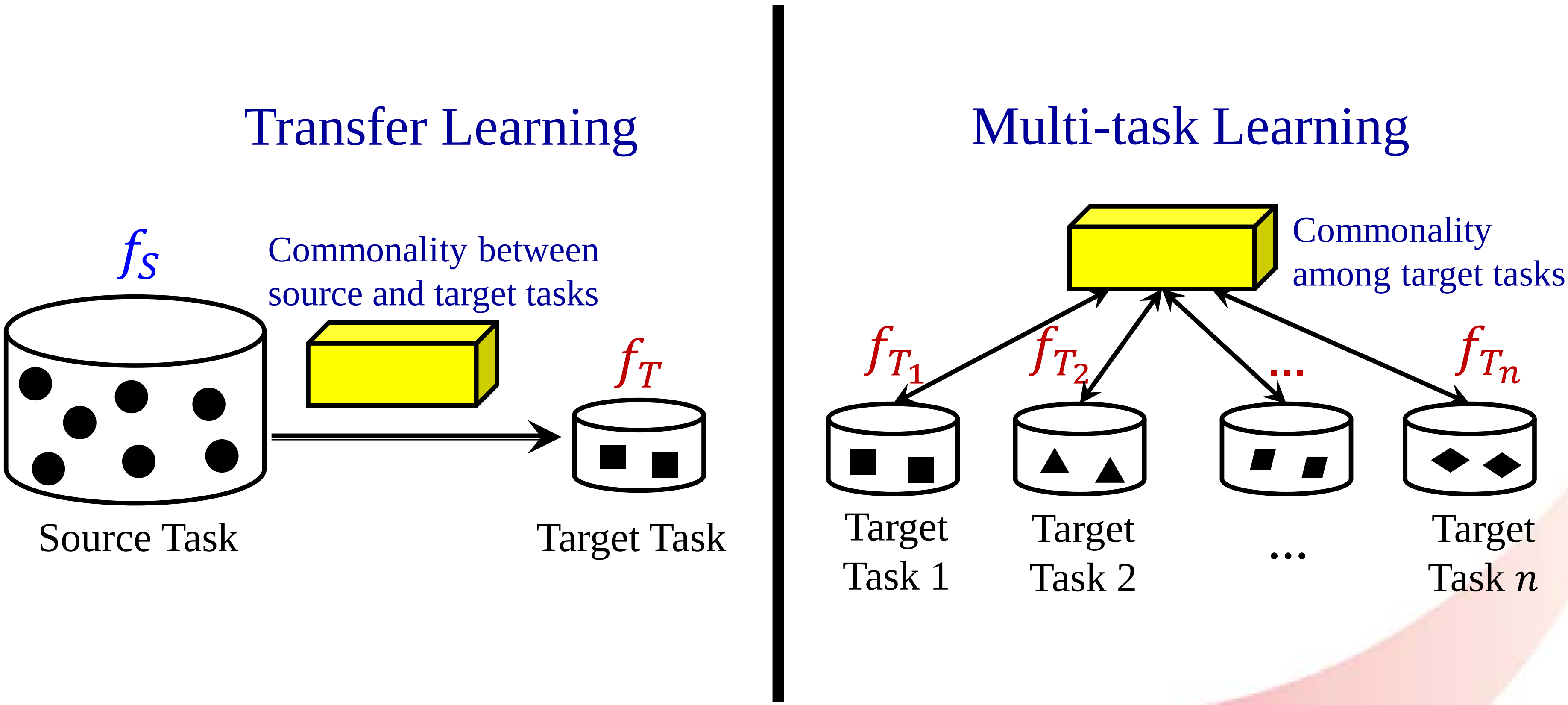
TL+ Semi-supervised Learning



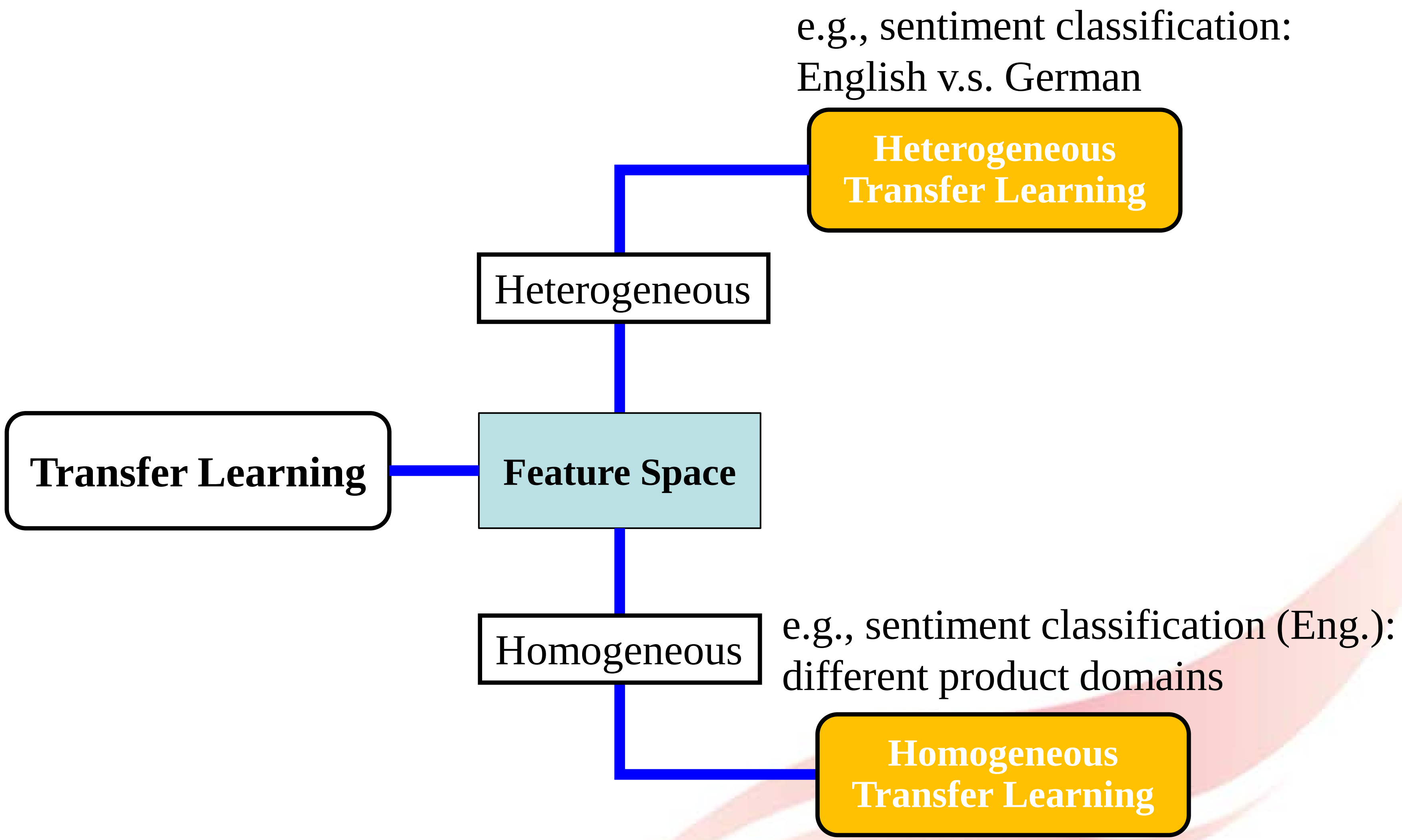
TL+ Active Learning



TL v.s. Multi-task Learning



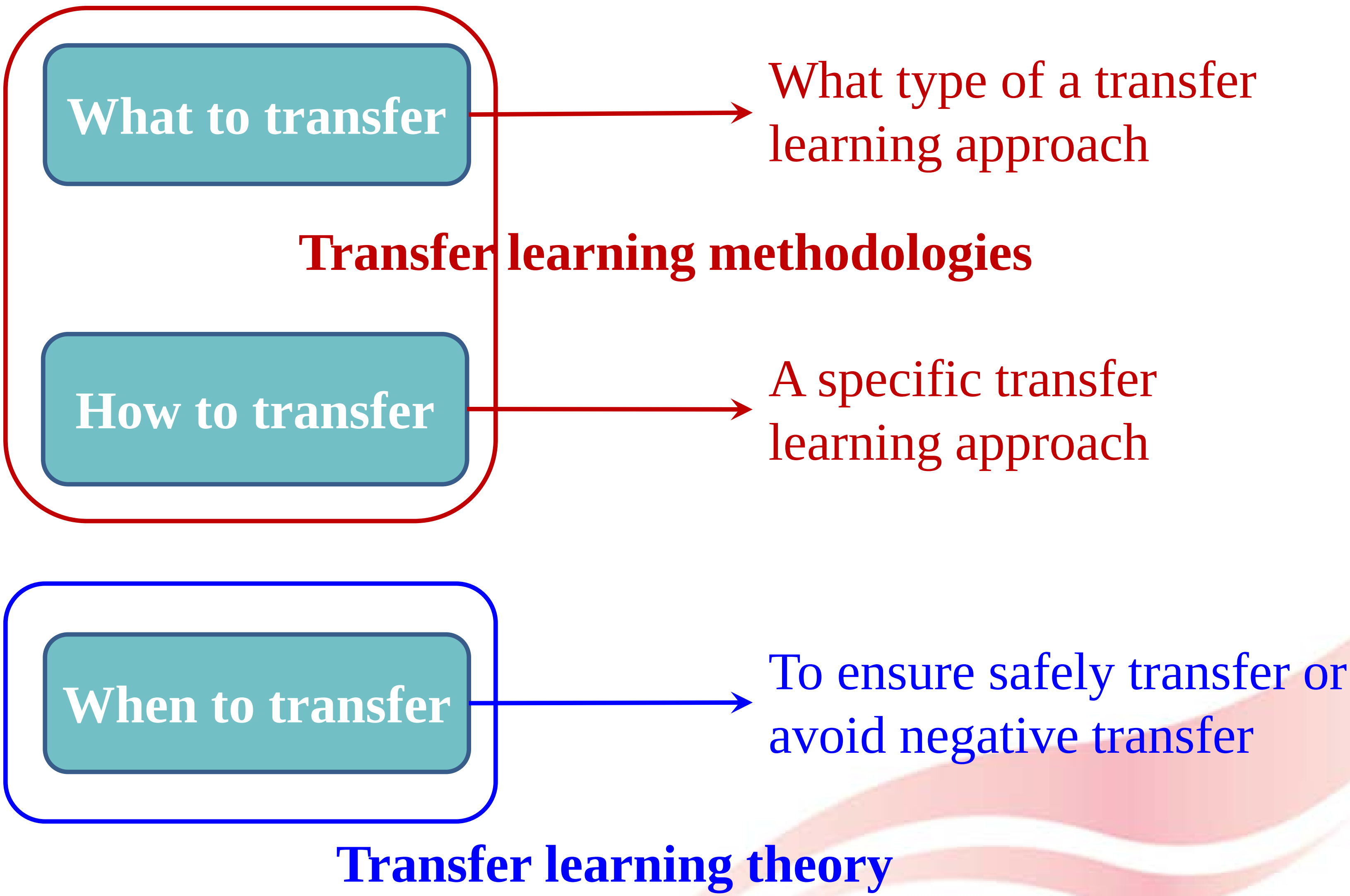
Different TL Settings



Research Issues in TL

What to transfer	What knowledge across domains/tasks can be transferred
How to transfer	Once what knowledge to be transferred is identified, how to encode the knowledge into a learning algorithm to transfer
When to transfer	In which situations, transfer learning can be safely performed

Research Issues in TL (cont.)



Transfer Learning Approaches

Based on “what to transfer”

Instance-based Approaches	**Feature-based Approaches**
Parameter-based Approaches	**Relational Approaches**

TL Approaches (cont.)

Instance-based Approaches — Knowledge to be transferred corresponds to the weights attached to source instances

Feature-based Approaches — Knowledge to be transferred corresponds to be the learned features across domains

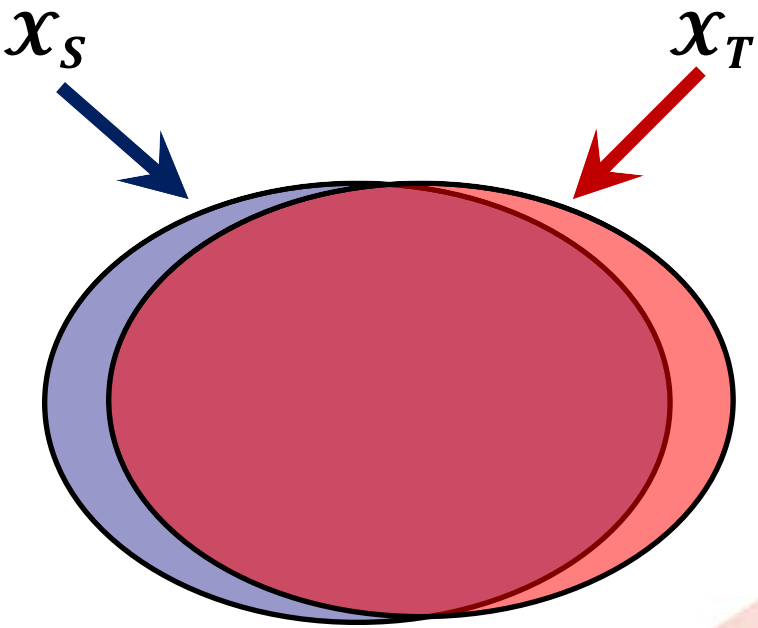
Parameter-based Approaches — Knowledge to be transferred is embedded in part of the source models

Relational Approaches — Knowledge to be transferred corresponds to the rules specifying the relations between entity in the source

Instance-based TL Approaches

General Assumption

Source and target domains have a lot of overlapping features (domains share the same/similar support)



Instance-based TL Approaches

Case I

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

$P_S(y|x) = P_T(y|x)$

$P_S(x) \neq P_T(x)$

Case II

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

$P_S(y|x) \neq P_T(y|x)$

Instance-based Approaches: Case I

Given a target task, based on the learning framework of empirical risk minimization

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

Instance-based Approaches: Case I (cont.)

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

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

$$= \arg\min \mathbb{E}_{(x,y)\sim P_S}\left[\frac{P_T(x)\cancel{P_T(y|x)}}{P_S(x)\cancel{P_S(y|x)}} l(x,y,\theta)\right]$$

$$\boxed{= \arg\min \mathbb{E}_{(x,y)\sim P_S}\left[\frac{P_T(x)}{P_S(x)} l(x,y,\theta)\right]}$$

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

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

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]

Classic Approaches

- Modeling a sampling selection biased process [Zadrozny, ICML-04]
  - Assume the difference between $P_S(x)$ and $P_T(x)$ is caused by a biased sample selection process
- Approximate $\beta(x)$ by a linear combination of some base functions [Sugiyama *etal*., NIPS-07, Kanamori *etal*., JMLR-09]
  - $\beta(x) = \sum_{\ell=1}^{b} \alpha_\ell \psi_\ell(x)$, where the coefficients $\alpha_\ell' s$ are to be learned
- Kernel mean matching (KMM) [Huang etal., NIPS-06]
  - Kernel embedding of distributions: Maximum Mean Discrepancy (MMD)
    [Alex Smola, Arthur Gretton and Kenji Kukumizu, ICML-08 tutorial]

Instance-based Approaches: Case II

- Assumption: $P_S(y|x) \neq P_T(y|x)$
- Recall:

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

$$\neq \frac{P_T(x)}{P_S(x)}$$

- Intuitive idea: 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

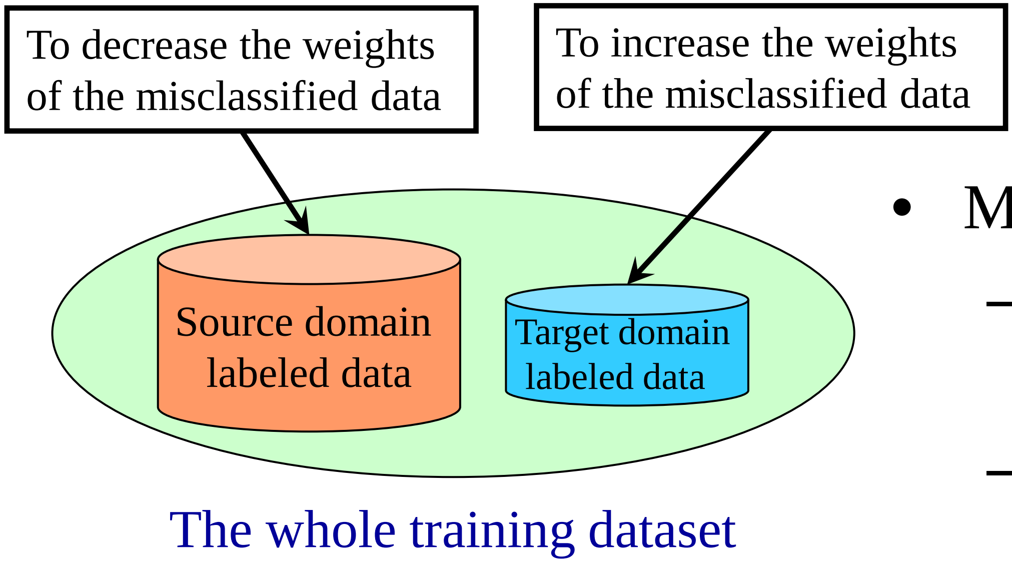
TraAdaBoost [Dai *etal* ICML-07]

- A boosting style approach to transfer learning
- High-level idea:
  - Use the same strategy as a standard boosting approach to update the weights of target domain data
  - Use a new mechanism to decrease the weights of misclassified source domain data

Boosting Procedure: Review

1. Initially, all training examples are assigned equal weights, so that they are equally likely to be chosen for training. A sample is drawn uniformly to obtain a new training set.
2. A classifier is induced from the training set, and used to classify all the examples in the original training set
3. The weights of the training examples are updated at the end of each boosting round
   - Records that are wrongly classified will have their weights increased
   - Records that are classified correctly will have their weights decreased
4. Repeat Step 2 and 3 until the stopping condition is met
5. Finally, the ensemble is obtained by aggregating the base classifiers obtained from each boosting round

TrAdaBoost

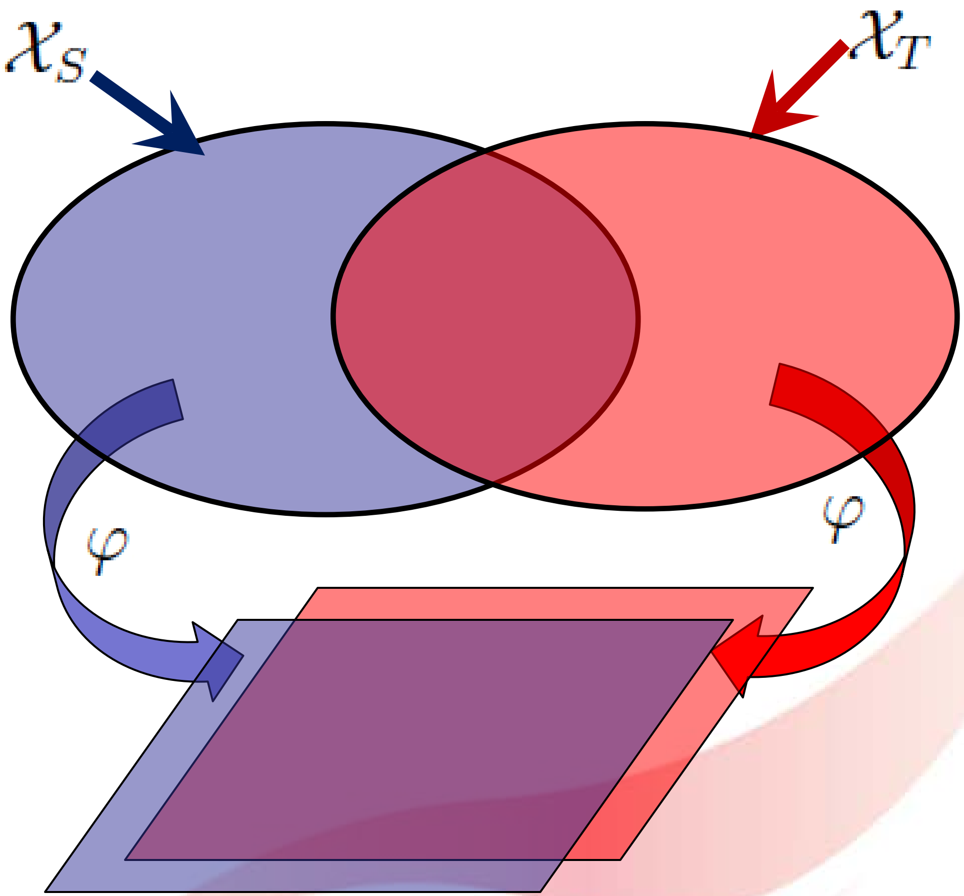


- Misclassified instances:
  - increase the weights of the misclassified target data
  - decrease the weights of the misclassified source data

TrAdaBoost is build on top of AdaBoost

Feature-based TL Approaches

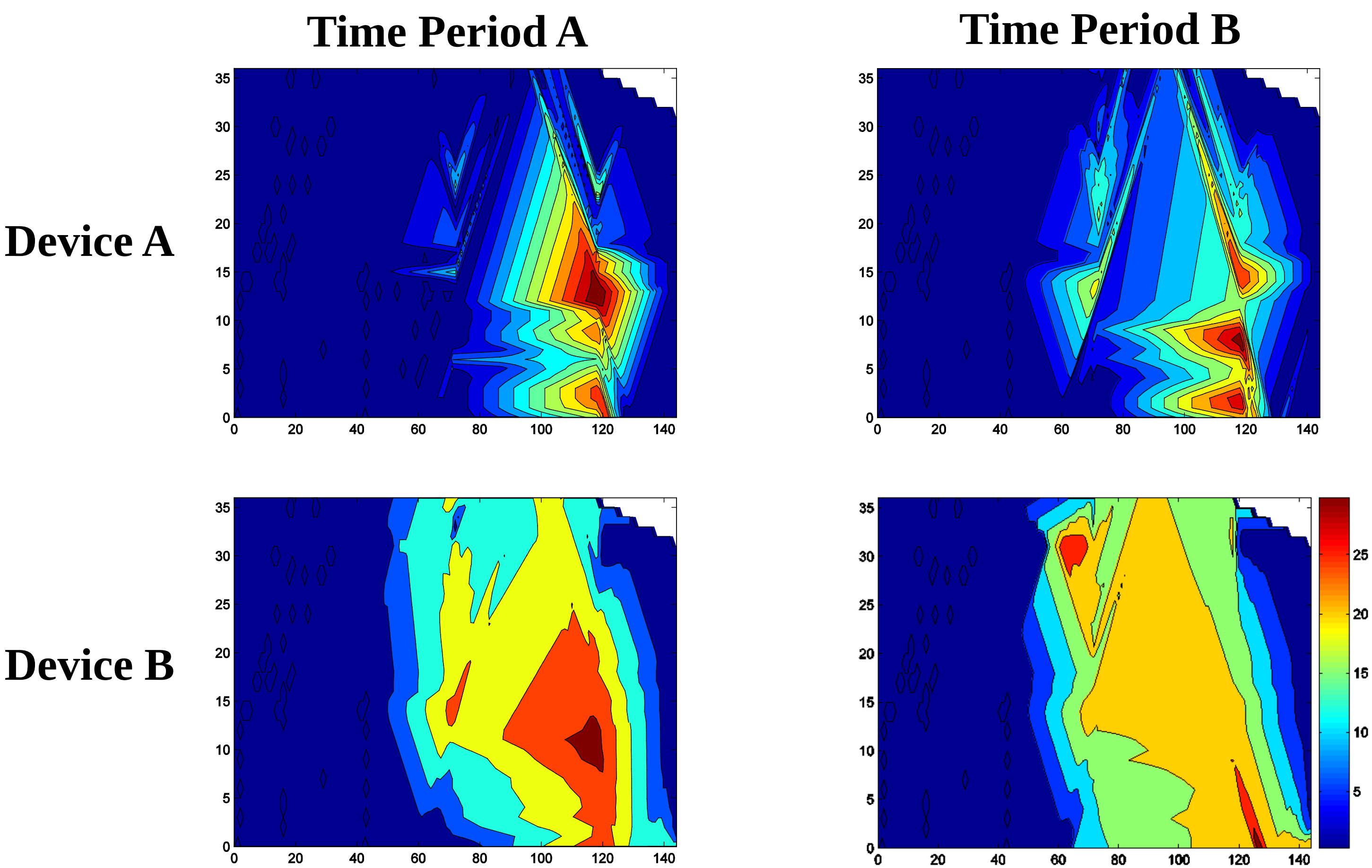
When source and target domains only have some overlapping features. (lots of features only have support in either the source or the target domain)



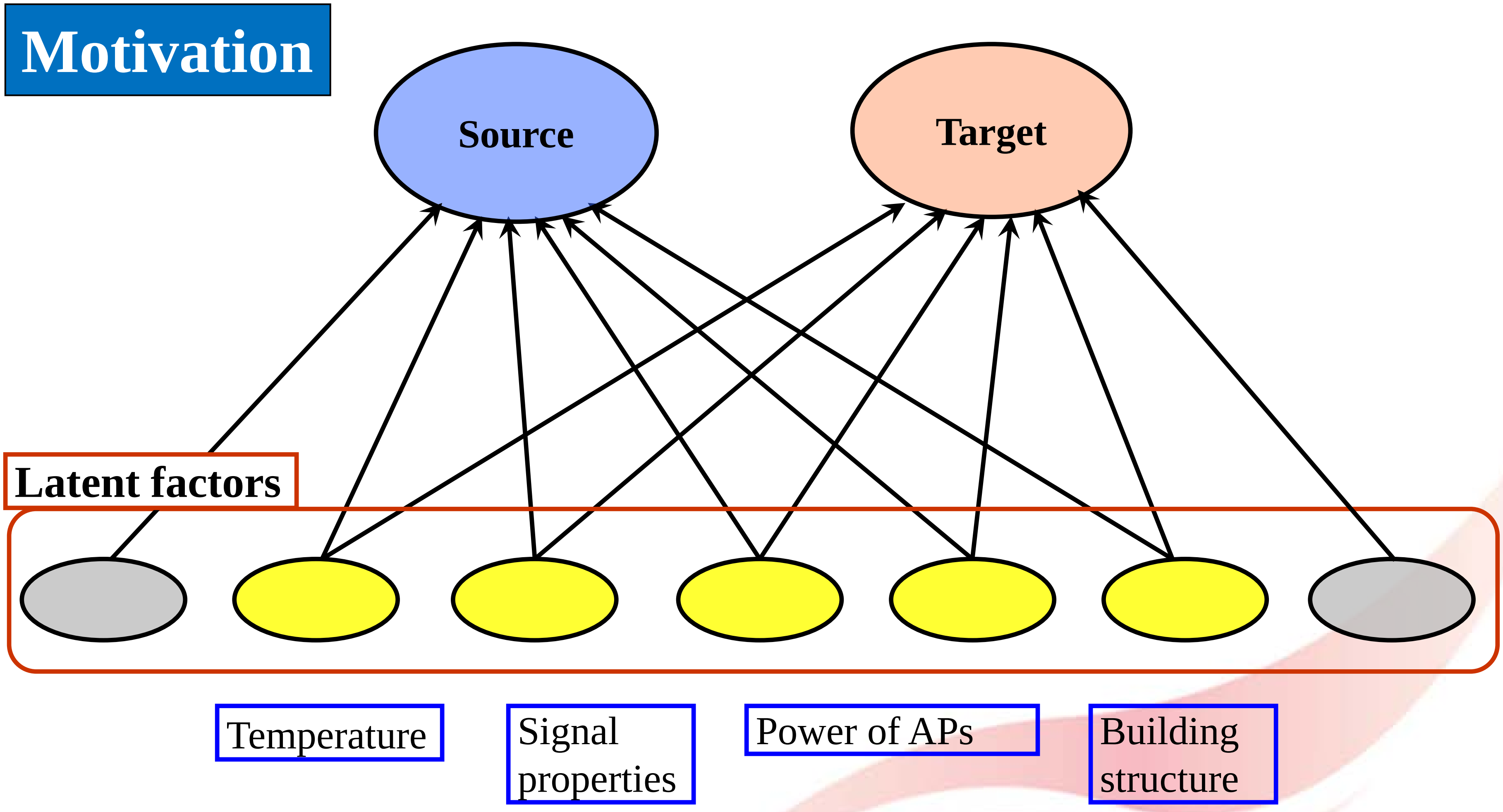
General Feature-based TL Approaches

- General approaches to learning the transformation
  - Learning features by minimizing distance between distributions
  - Learning universal features via self-taught learning

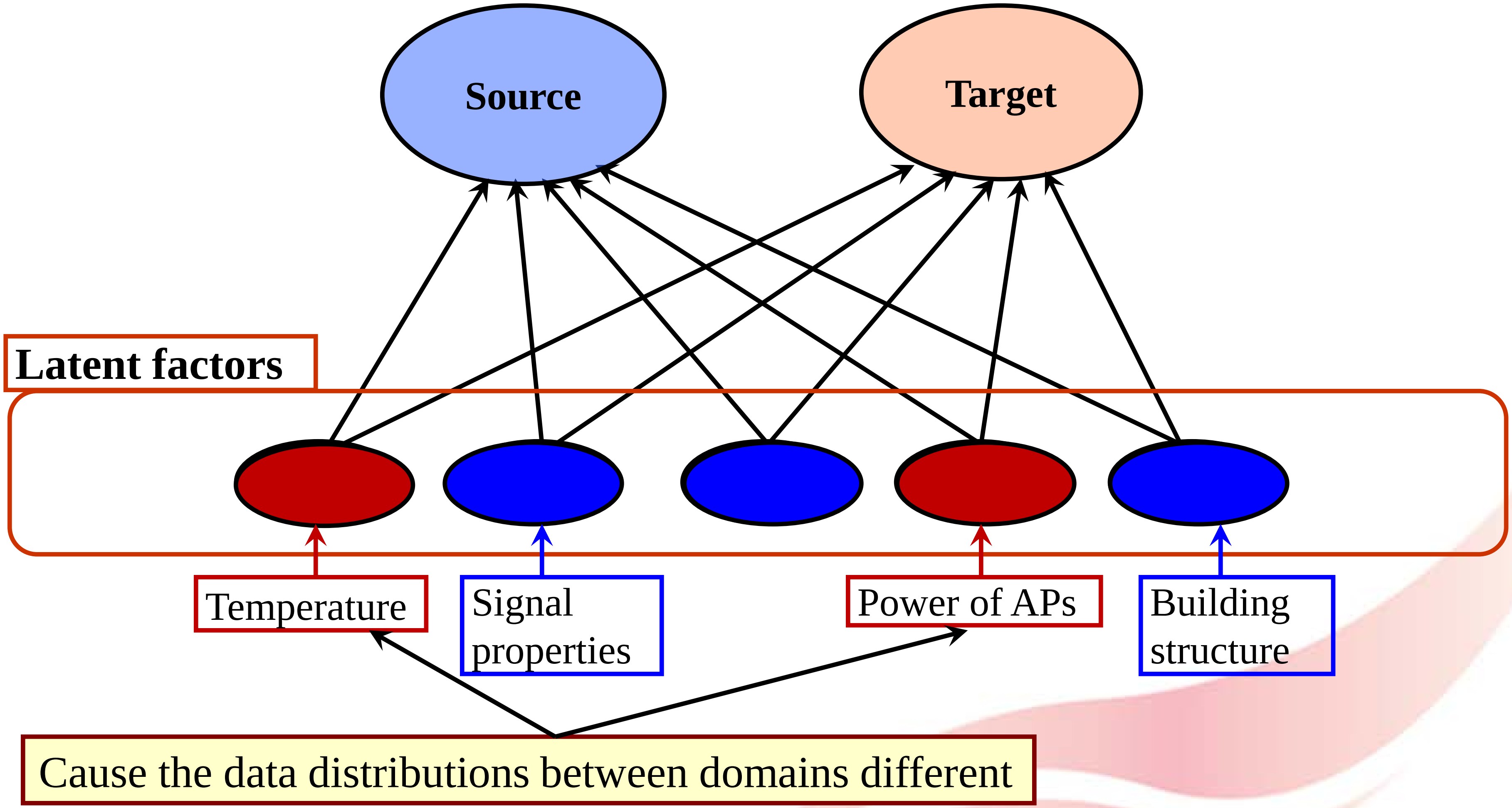
An Example



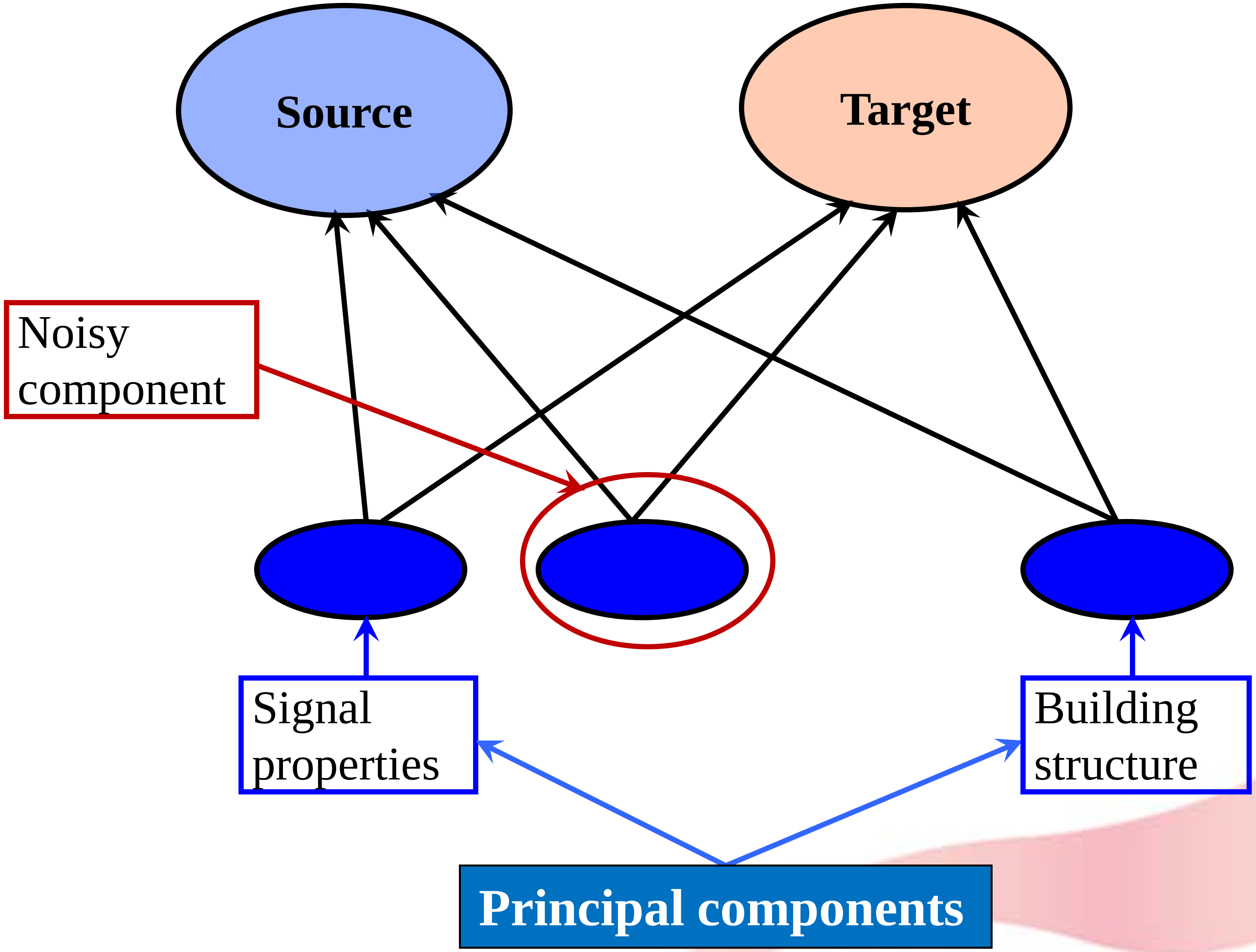
Transfer Component Analysis (TCA)
[Pan *etal*., IJCAI-09, TNN-11]



TCA (cont.)

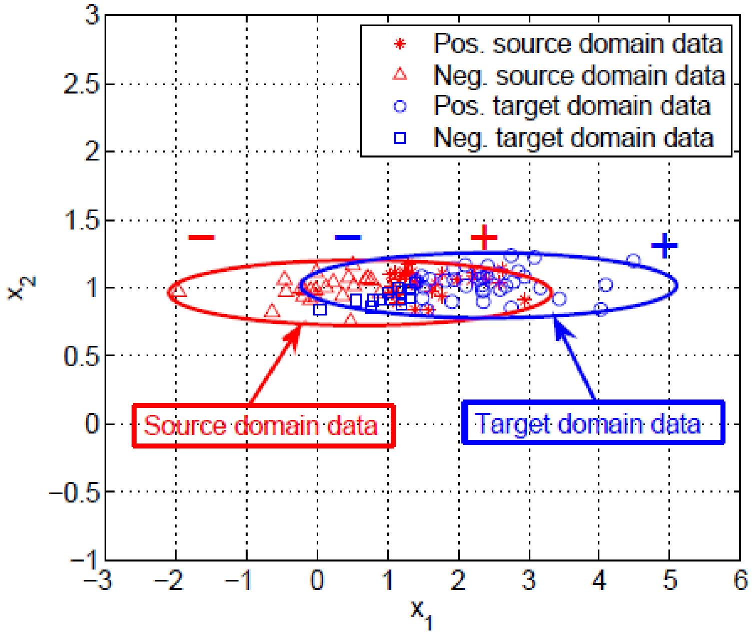

Source
Target
Latent factors
Temperature
Signal properties
Power of APs
Building structure
Cause the data distributions between domains different

TCA (cont.)


Source
Target
Noisy component
Signal properties
Building structure
Principal components

TCA (cont.)

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



TCA (cont.)

- **Main idea:** the learned φ should map the source domain and target domain data to a latent space spanned by the factors that reduce domain distance as well as preserve data structure
- **High level optimization problem**

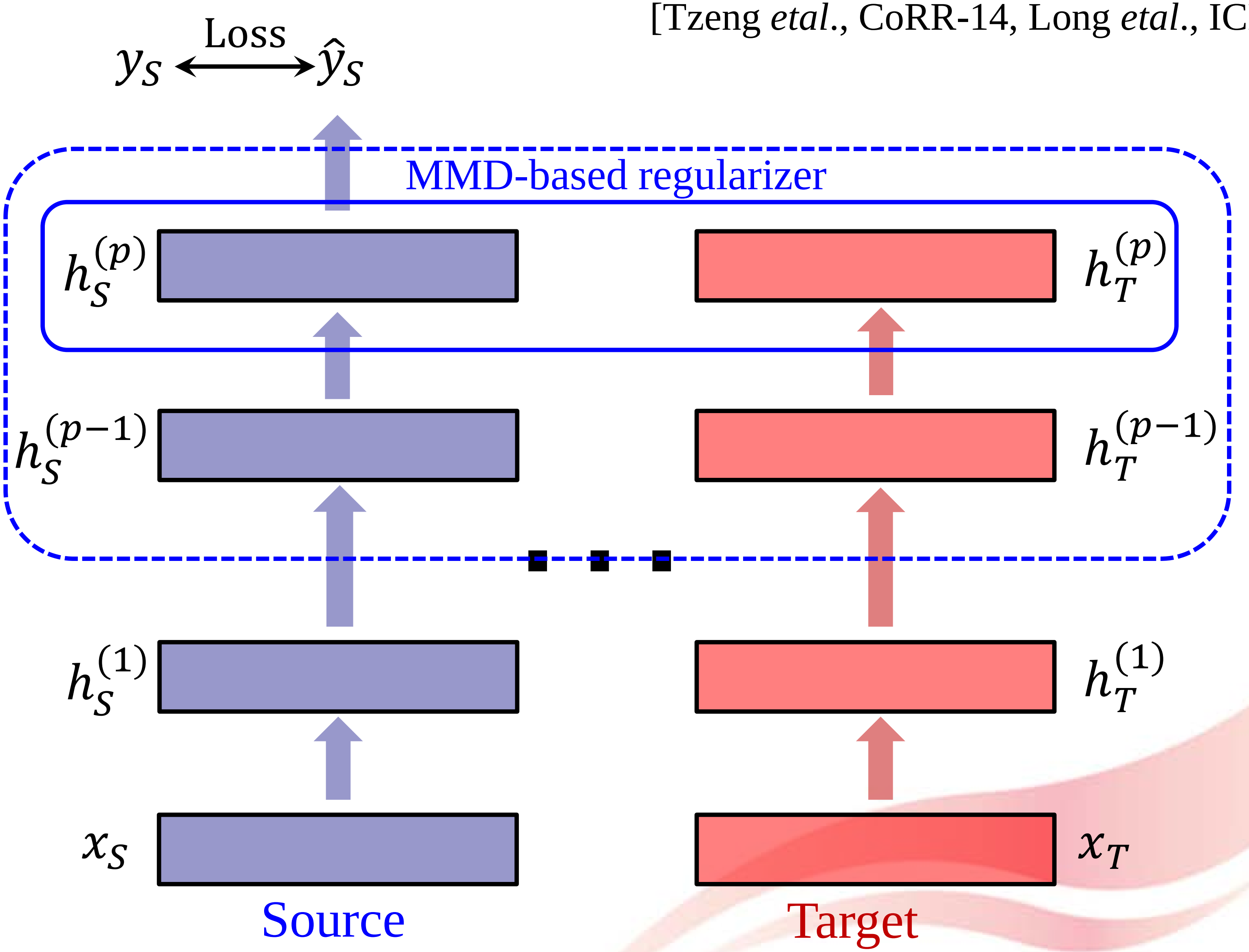
$$\min_{\varphi} \; \boxed{\mathrm{Dist}\big(\varphi(X_S), \varphi(X_T)\big)} + \lambda\Omega(\varphi)$$
$$\text{s.t.} \; \boxed{\text{constraints on } \varphi(X_S) \text{ and } \varphi(X_T)}$$

Preserve data variance

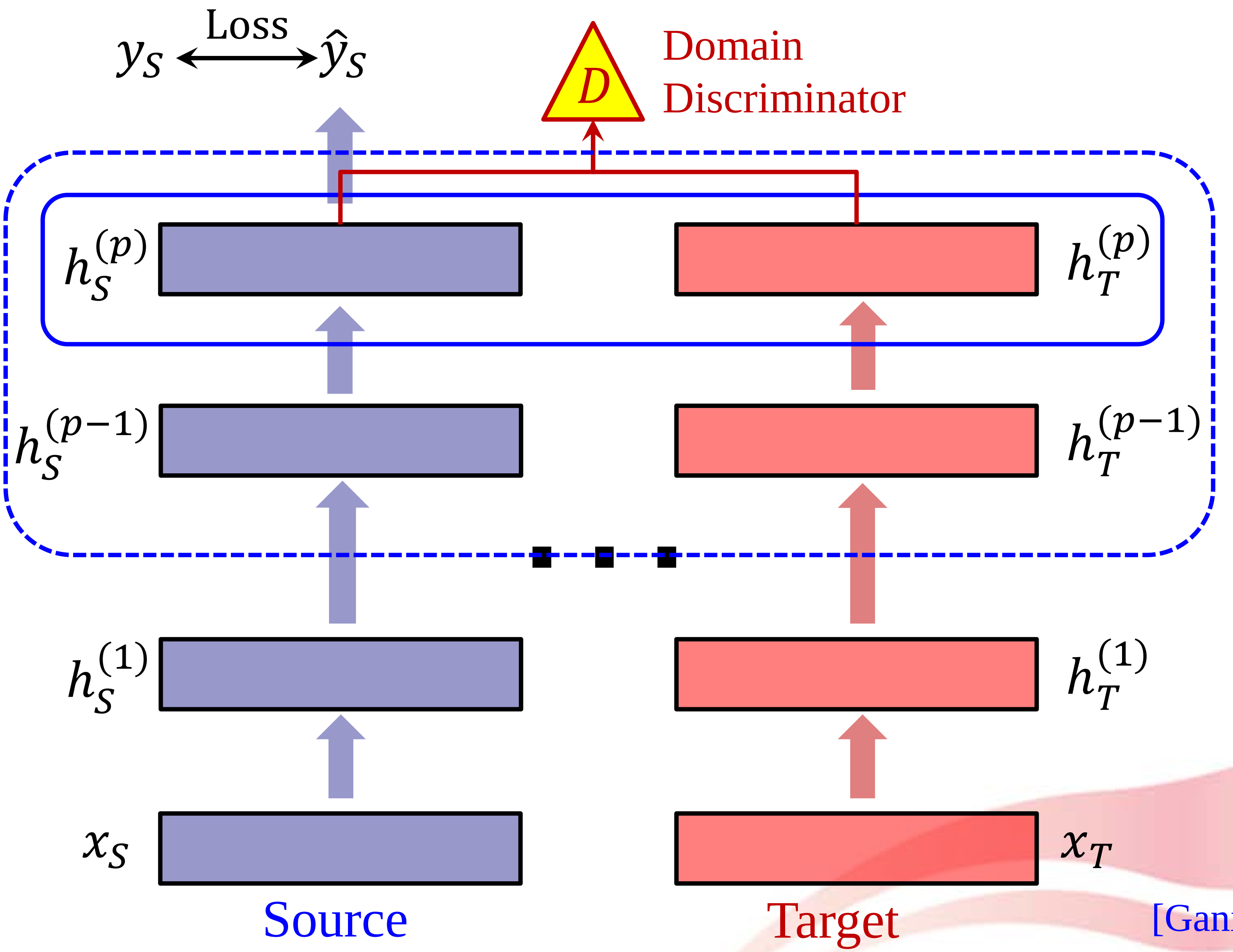
Maximum Mean Discrepancy (MMD)

Extension to Deep Architecture

[Tzeng *etal*., CoRR-14, Long *etal*., ICML-15]



Domain Adversarial Training



Objective 1: learn hidden features to obtain low loss: minimization

Objective 2: learn hidden features to confuse domain discriminator: maximization

[Ganin *etal*., JMLR-16]

General Feature-based TL Approaches

- General approaches to learning the transformation
  - Learning features by minimizing distance between distributions
  - <u>Learning universal features via self-taught learning</u>

Self-taught Feature Learning

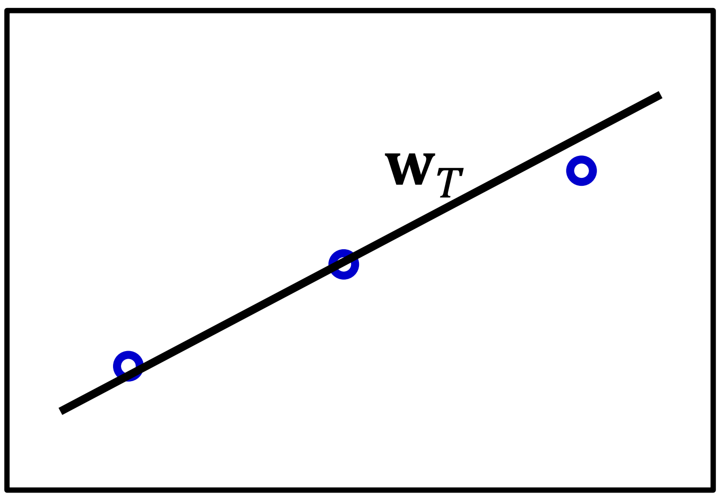
- **Motivation:**
  - There exist some high-level features that can help the target learning task even only a few labeled data are given
  - High-level features can be learned in advance from auxiliary tasks or domains
- **General steps:**
  - Learn high-level features from a lot of unlabeled data (which can come from various known or unknown domains)
  - Use the learned high-level features to represent the data of the target task
  - Training models with the new representation

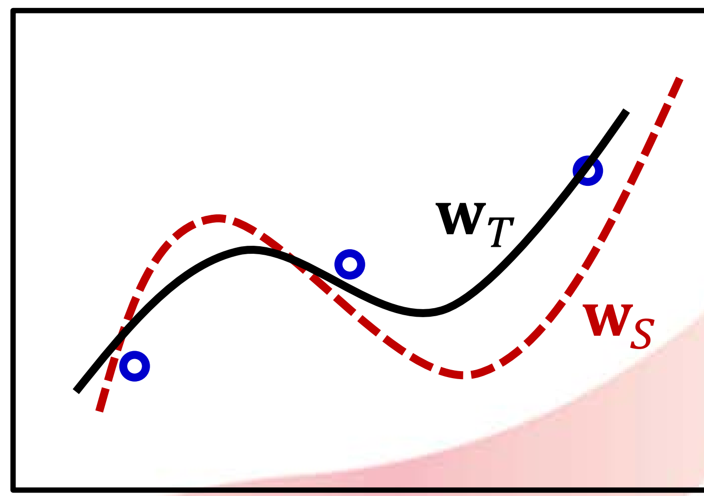
Self-taught Feature Learning (cont.)

- How to learn universal high-level features
  - Sparse Coding [Raina etal., 2007]
  - Autoencoder [Glorot *etal*., 2011]
  - Other deep learning models, e.g., CNNs

Parameter-based Approaches

- **Motivation:** A well-trained source model $\mathbf{w}_S$ has captured a lot of structure from data. If two tasks are related, this structure can be transferred to learn a more precise target model $\mathbf{w}_T$ with a few labeled data in the target domain





Parameter-based TL Approaches (cont.)

- Assumption: if tasks are related, they may share similar parameter vectors

Common part

$$\mathbf{w}_S = \boxed{\mathbf{w}_0} + \mathbf{v}_S$$

$$\mathbf{w}_T = \boxed{\mathbf{w}_0} + \mathbf{v}_T$$

Specific part for individual task

Optimized by using all the data

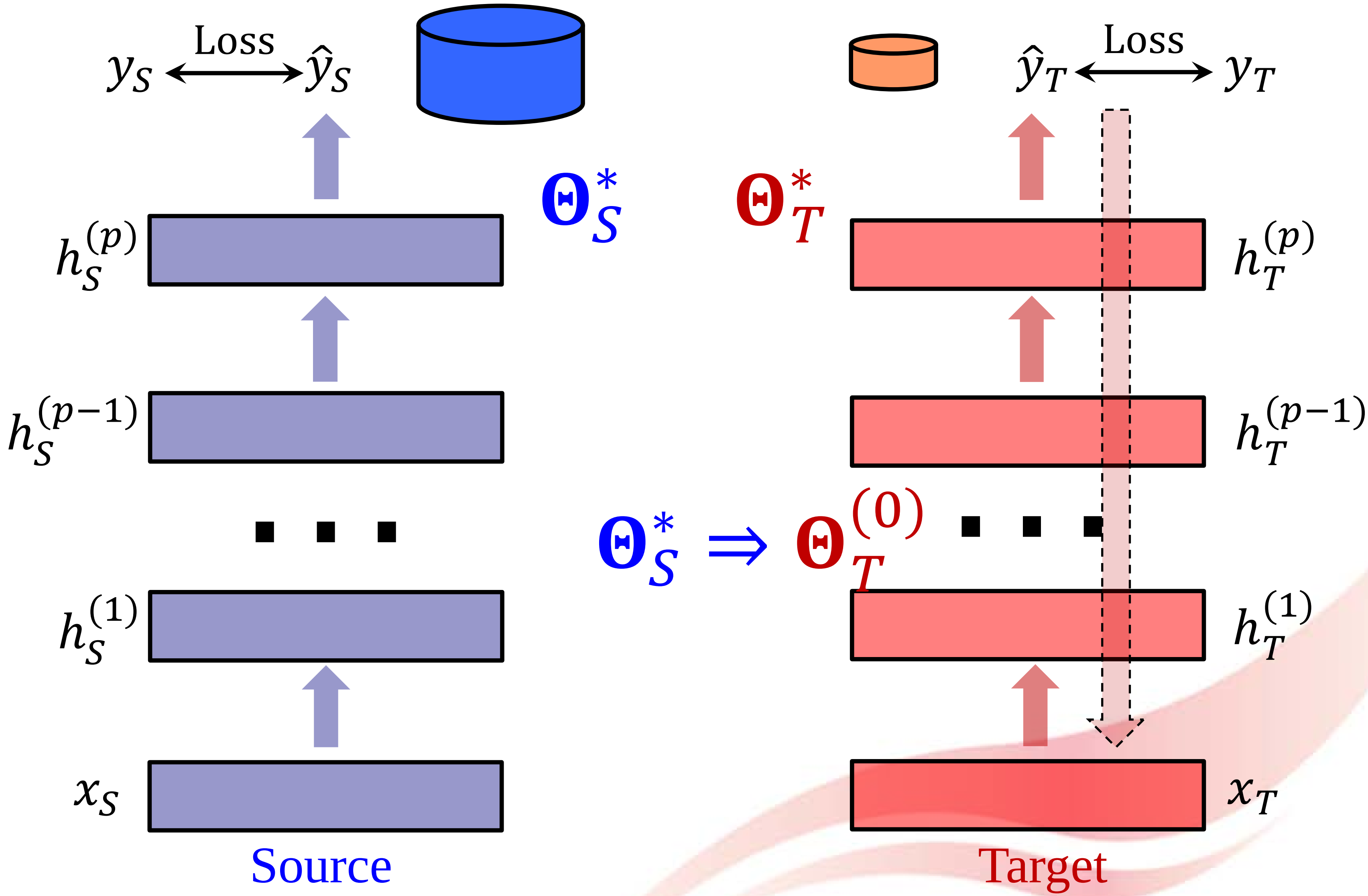
$$\min_{\mathbf{v}_S,\mathbf{v}_T,\mathbf{w}_0} \sum_{t\in\{S,T\}} \frac{\gamma_t}{n_t} \sum_{i=1}^{n_t} l(x_{t_i}, y_{t_i}; \mathbf{w}_t) + \lambda_1 \boxed{(\|\mathbf{v}_S\|_2^2 + \|\mathbf{v}_T\|_2^2)} + \lambda_2 \|\boxed{\mathbf{w}_0}\|_2^2$$

Optimized by using the data of individual task, respectively

[Evgeniou and Pontil, KDD-04]

[Yosinski *etal.*, NIPS-14]

In the Context of Deep Learning

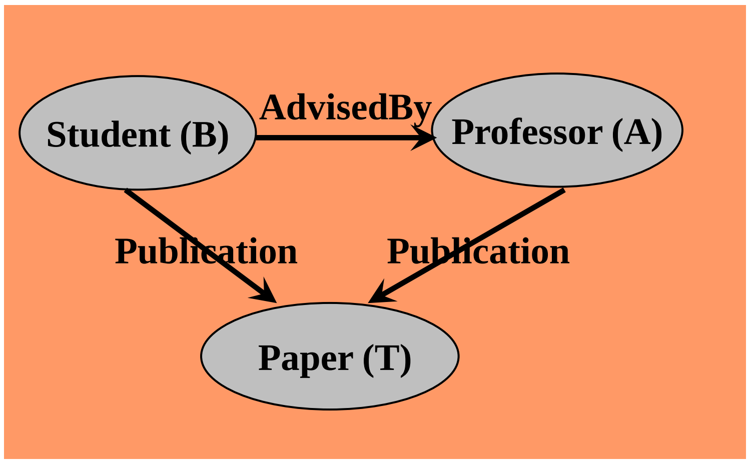


Relational TL Approaches

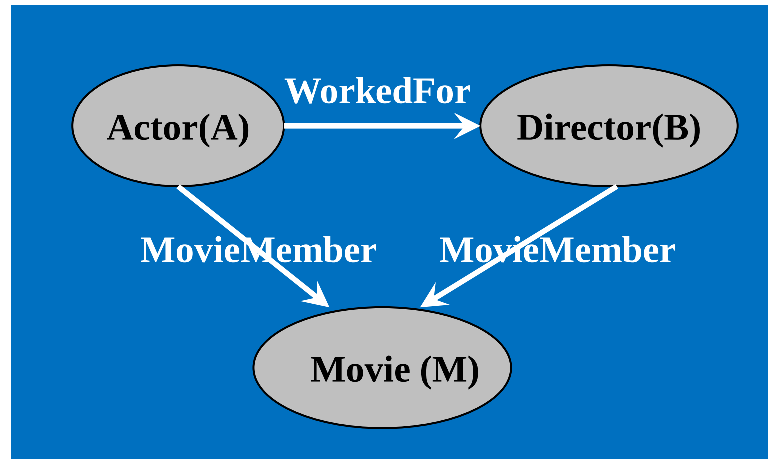
- **Motivation:** If two relational domains (non-i.i.d) are related, they may share some similar relations among objects. These relations can be used for knowledge transfer across domains

Motivating Example

Academic domain (source)



Movie domain (target)



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

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

$\mathrm{P1}(x, y) \land \mathrm{P2}(x, z) \Rightarrow \mathrm{P2}(y, z)$

Summary

In data level

Instance-based Approaches	Knowledge to be transferred corresponds to the weights attached to source instances
Feature-based Approaches	Knowledge to be transferred corresponds to be the learned features across domains
Parameter-based Approaches	Knowledge to be transferred is embedded in part of the source models
Relational Approaches	Knowledge to be transferred corresponds to the rules specifying the relations between entity in the source

In model level

Thank You!

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]
- [Pan, Transfer Learning, Chapter 21, Data Classification: Algorithms and Applications 2014]
- [Quionero-Candela, *etal*, Data Shift in Machine Learning, MIT Press 2009]
- [Zadrozny, Learning and Evaluating Classifiers under Sample Selection Bias, ICML 2004]
- [Sugiyama *etal*., Direct Importance Estimation with Model Selection and Its Application to Covariate Shift Adaptation, NIPS 2007]

Reference (cont.)

- [Kanamori *etal*., A Least-squares Approach to Direct Importance Estimation, JMLR 2009]
- [Huang *etal*., Correcting Sample Selection Bias by Unlabeled Data, NIPS 2006]
- [Dai *etal*., Boosting for Transfer Learning, ICML 2007]
- [Pan *etal*., Transfer Learning via Dimensionality Reduction, AAAI 2008]
- [Pan *etal*., Domain Adaptation via Transfer Component Analysis, IJCAI 2009, TNN 2011]
- [Tzeng *etal*., Deep Domain Confusion: Maximizing for Domain Invariance, CoRR 2014]
- [Long *etal*., Learning Transferable Features with Deep Adaptation Networks, ICML 2015]
- [Ganin *etal*., Domaina Adversarial Training of Neural Networks, JMLR 2016]

Reference (cont.)

- [Raina *etal*., Self-taught Learning: Transfer Learning from Unlabeled Data, ICML 2007]
- [Yosinski *etal*., How Transferable Are Features in Deep Neural Networks? NIPS 2014]
- [Davis and Domingos, Deep Transfer vis Second-order Markov Logic, ICML 2009]