TRANSFER LEARNING

Wenbao Li

Outline

- **Transfer Learning**
- **TrAdaboost**
- **Self-taught Learning**
- **Self-taught Clustering**

Motivation

When the distribution changes:

- ● new labeled data is short,
- ● expensive to recollect the needed training data
- ● impossible or time consuming to rebuild models

At the same time:

- ● a waste to drop old data and its learning model.

◇ *How to extract knowledge learnt from related domains to help learning in a target domain with a few labeled data?*

◇ *How to extract knowledge learnt from related domains to speed up learning in a target domain?*

✌ **Transfer learning techniques may help!**

Transfer Learning?

Transfer Learning (TL):

The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks (in new domains)

It is motivated by human learning. People can often transfer knowledge learnt previously to novel situations

✓ Chess → Checkers

✓ Mathematics → Computer Science

✓ Table Tennis → Tennis

Transfer Learning

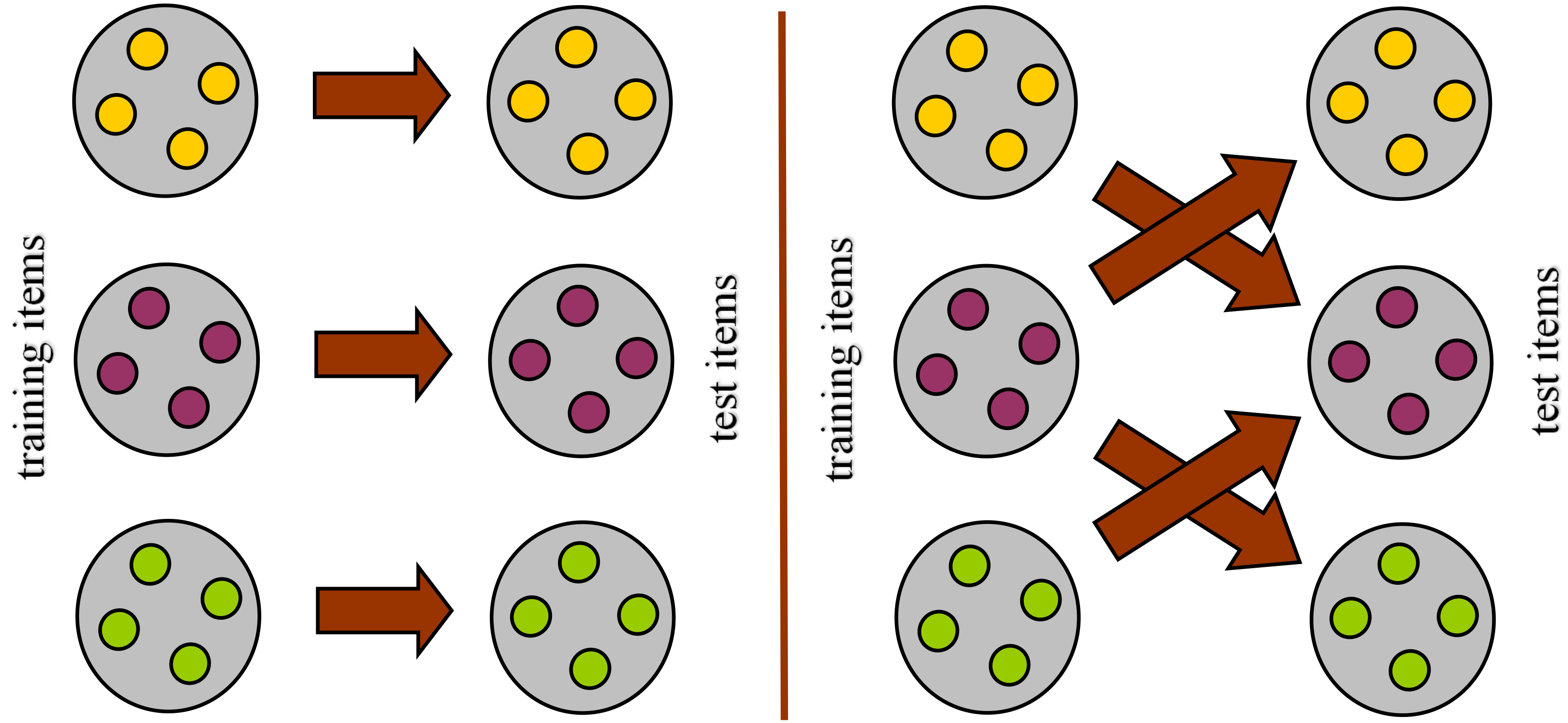
- Traditional Machine Learning vs. Transfer Learning
- Settings of Transfer Learning
- Approaches to Transfer Learning
- Negative Transfer
- Conclusion

Traditional ML vs. TL

(P. Langley 06)

Traditional ML in multiple domains

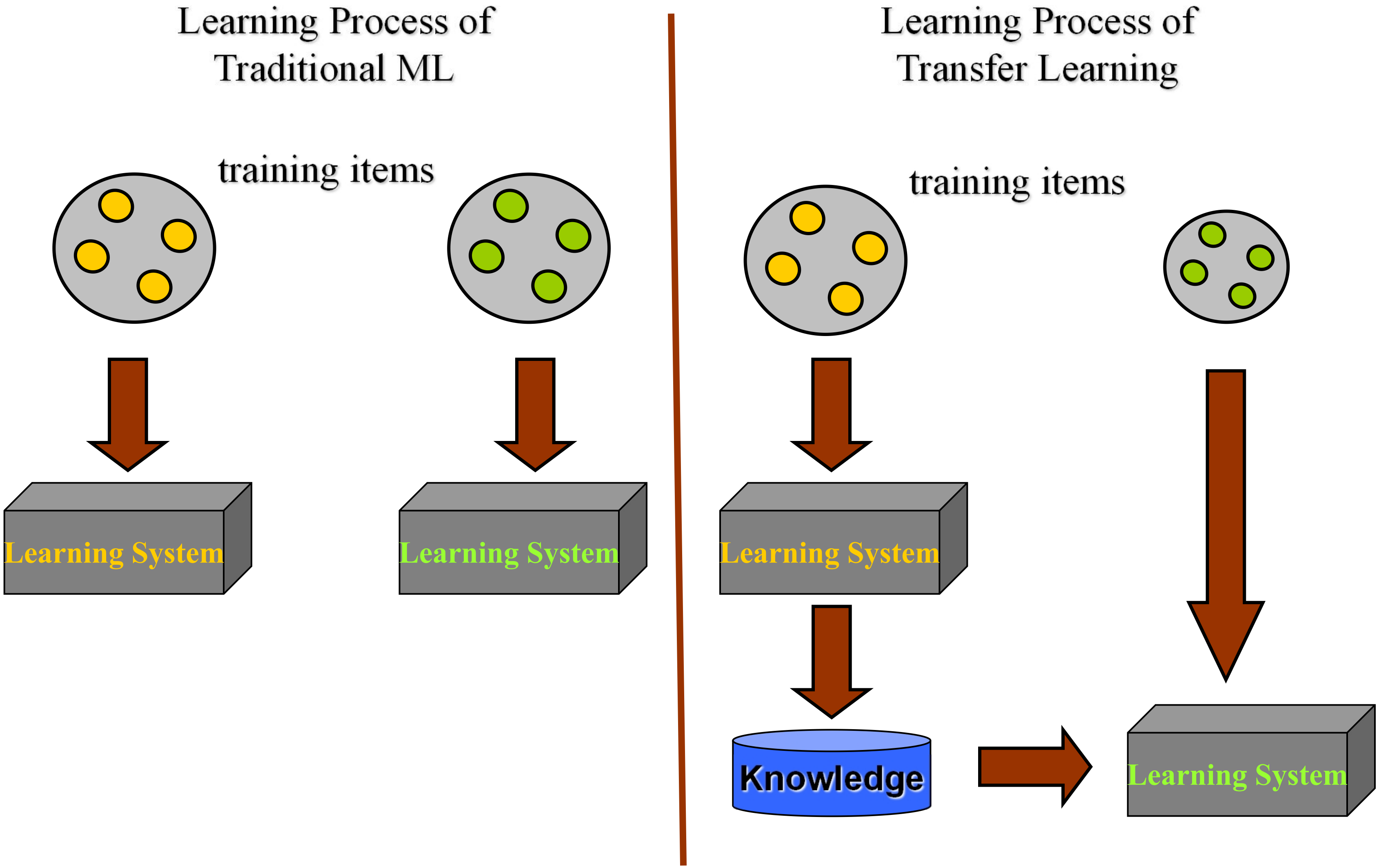
Transfer of learning across domains



Humans can learn in many domains.

Humans can also transfer from one domain to other domains.

Traditional ML vs. TL



Learning Process of Traditional ML

training items

Learning System

Learning System

Learning Process of Transfer Learning

training items

Learning System

Knowledge

Learning System

Notation

Domain:

It consists of two components: A feature space $\mathcal{X}$**, a marginal distribution**

$\mathcal{P}(X)$, where $X = \{x_1, x_2, ..., x_n\} \in \mathcal{X}$

In general, if two domains are different, then they may have different feature spaces or different marginal distributions.

Task:

Given a specific domain and label space $\mathcal{Y}$ **, for each** x_i **in the domain, to predict its corresponding label** y_i, where $y_i \in \mathcal{Y}$

In general, if two tasks are different, then they may have different label spaces or different conditional distributions

$\mathcal{P}(Y|X)$, where $Y = \{y_1, ..., y_n\}$ and $y_i \in \mathcal{Y}$

Notation

For simplicity, we only consider at most two domains and two tasks.

Source domain:

$$\mathcal{P}(X_S), \text{ where } X_S = \{x_{S_1}, x_{S_2}, ..., x_{S_{n_S}}\} \in \mathcal{X}_S$$

Task in the source domain:

$$\mathcal{P}(Y_S|X_S), \text{ where } Y_S = \{y_{S_1}, y_{S_2}, ..., y_{S_{n_S}}\} \text{ and } y_{S_i} \in \mathcal{Y}_S$$

Target domain:

$$\mathcal{P}(X_T), \text{ where } X_T = \{x_{T_1}, x_{T_2}, ..., x_{T_{n_T}}\} \in \mathcal{X}_T$$

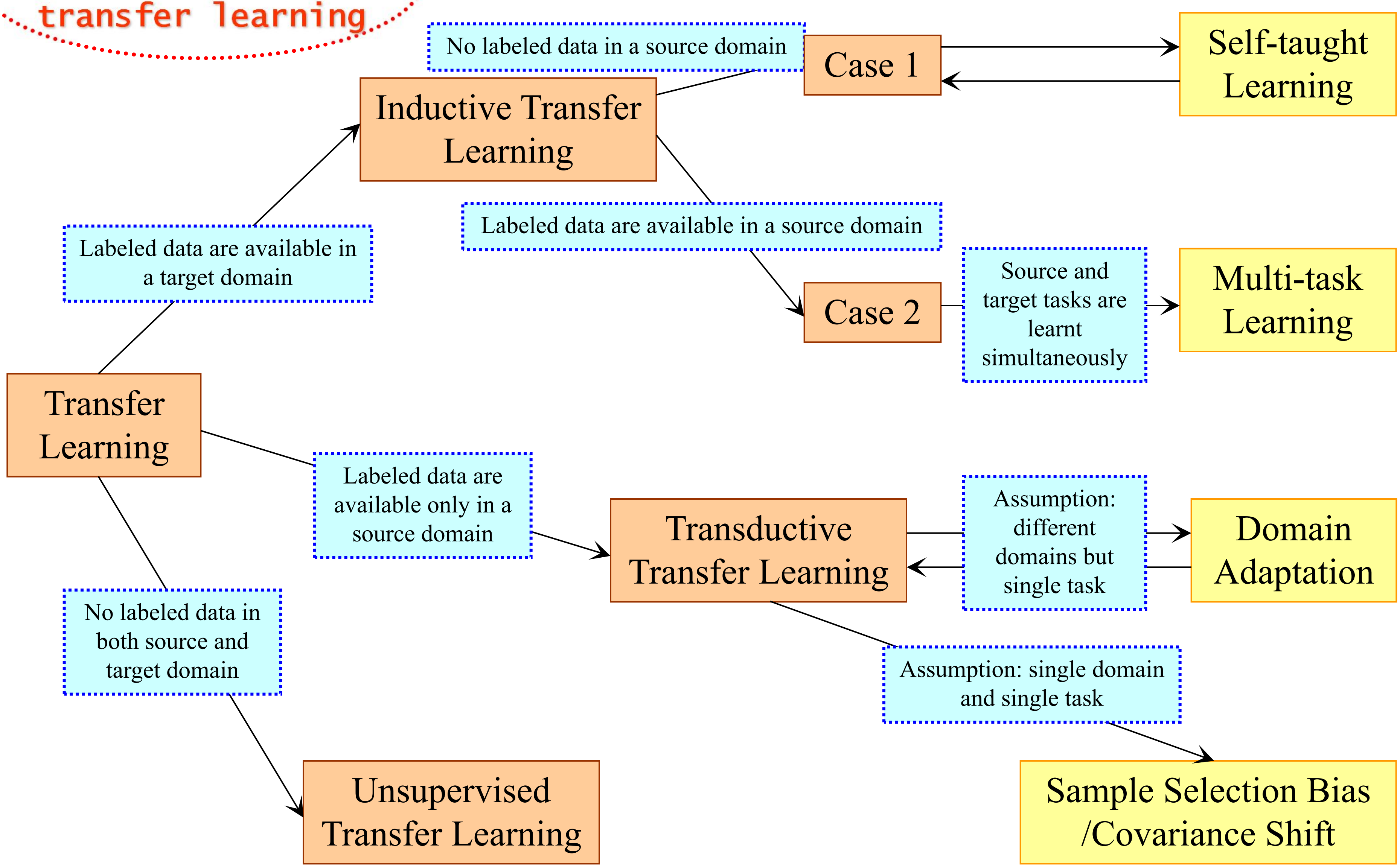
Task in the target domain

$$\mathcal{P}(Y_T|X_T), \text{ where } Y_T = \{y_{T_1}, y_{T_2}, ..., y_{T_{n_T}}\} \text{ and } y_{T_i} \in \mathcal{Y}_T$$

Settings of Transfer Learning

Transfer learning settings	**Labeled data in a source domain**	**Labeled data in a target domain**	**Tasks**
Inductive Transfer Learning	×	√	Classification Regression …
	√	√	
Transductive Transfer Learning	√	×	Classification Regression …
Unsupervised Transfer Learning	×	×	Clustering …

An overview of various settings of transfer learning



Transfer Learning

Labeled data are available in a target domain

Inductive Transfer Learning

No labeled data in a source domain

Case 1

Self-taught Learning

Labeled data are available in a source domain

Case 2

Source and target tasks are learnt simultaneously

Multi-task Learning

Labeled data are available only in a source domain

Transductive Transfer Learning

Assumption: different domains but single task

Domain Adaptation

Assumption: single domain and single task

Sample Selection Bias /Covariance Shift

No labeled data in both source and target domain

Unsupervised Transfer Learning

Approaches to Transfer Learning

-According to what to transfer

Transfer learning approaches	**Description**
Instance-transfer	*To re-weight some labeled data in a source domain for use in the target domain*
Feature-representation-transfer	Find a “good” feature representation that reduces difference between a source and a target domain or minimizes error of models
Model(Parameter)-transfer	Discover shared parameters or priors of models between a source domain and a target domain
Relational-knowledge-transfer	Build mapping of relational knowledge between a source domain and a target domain.

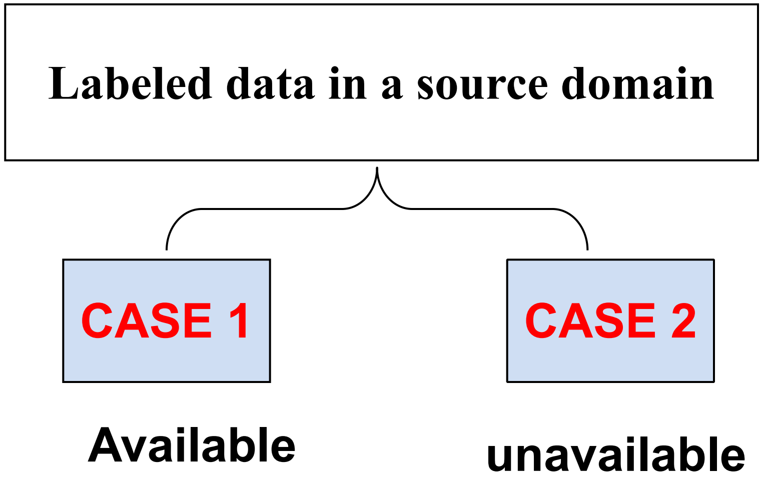
Approaches to Transfer Learning

	Inductive Transfer Learning	**Transductive Transfer Learning**	**Unsupervised Transfer Learning**
Instance-transfer	√	√	
Feature-representation-transfer	√	√	√
Model(Parameter)-transfer	√		
Relational-knowledge-transfer	√		

Inductive Transfer Learning

- aims to help improve the learning of the target predictive function

  $f_T(\ )$ in D_T using the knowledge in D_S and T_S ,where $T_S \neq T_T$



Inductive Transfer Learning

Instance-transfer Approaches

TrAdaBoost[Dai et al. ICML-07]

- Assumption: the source domain and target domain data use exactly the same features and labels.

- Motivation: Although the source domain data can not be reused directly, there are some parts of the data that can still be reused by re-weighting.

- Main Idea: Discriminatively adjust weighs of data in the source domain for use in the target domain.

- **Non-standard SVMs [Wu and Dietterich ICML-04]**
- **TrAdaBoost[Dai et al. ICML-07]**

Inductive Transfer Learning

Feature-representation-transfer Approaches

Supervised Feature Construction

Assumption: If t tasks are related to each other, then they may share some common features which can benefit for all tasks.

Input: t tasks, each of them has its own training data.

Output: Common features learnt across t tasks and t models for t tasks, respectively.

- [Argyriou et al. NIPS-06, NIPS-07]

Inductive Transfer Learning

Feature-representation-transfer Approaches

Unsupervised Feature Construction

Three steps:

1. Applying *sparse coding* [Lee et al. NIPS-07] algorithm to learn higher-level representation from unlabeled data in the source domain.

2. Transforming the target data to new representations by new bases learnt in the first step.

3. Traditional discriminative models can be applied on new representations of the target data with corresponding labels.

- **Self-taught learning[Raina et al. ICML-07]**

Unsupervised Feature Construction

[Raina et al. ICML-07]

Step1:

$$\min_{a,b} \sum_i \quad \|x_{S_i} - \textstyle\sum_j a^j_{S_i} b_j\|_2^2 + \beta\|a_{S_i}\|_1$$

$$s.t. \qquad \|b_j\|_2 \leq 1, \forall j \in 1, \ldots, s$$

Input: Source domain data $X_S = \{x_{S_i}\}$ and coefficient β

Output: New representations of the source domain data $A_S = \{\boldsymbol{a}_{S_i}\}$ and new bases $B = \{b_i\}$

Step2:

$$a^*_{T_i} = \arg\min_{a_{T_i}} \|x_{T_i} - \sum_j a^j_{T_i} b_j\|_2^2 + \beta\|a_{T_i}\|_1$$

Input: Target domain data $X_T = \{x_{T_i}\}$, coefficient β and bases $B = \{b_i\}$

Output: New representations of the target domain data $A_T = \{a_{T_i}\}$

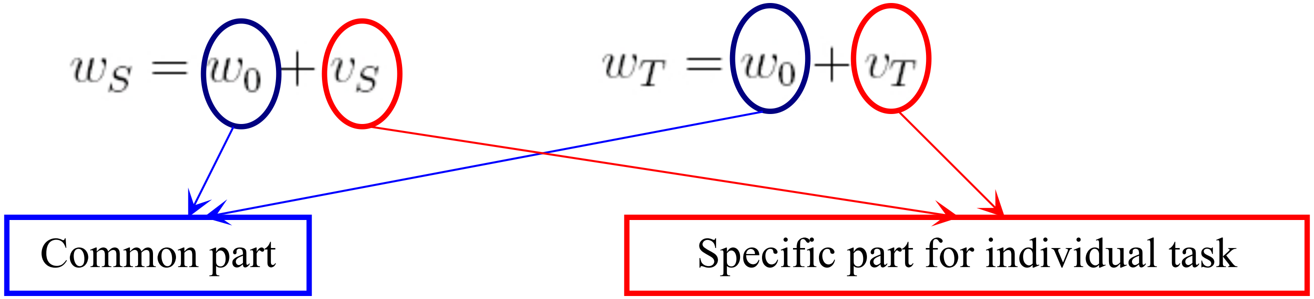
Inductive Transfer Learning

Model-transfer Approaches

Regularization-based Method [Evgeiou and Pontil, KDD-04]

Assumption: If t tasks are related to each other, then they may share some parameters among individual models.

Assume $f_t = w_t \cdot x$ be a hyper-plane for task , where $t \in \{T, S\}$ and



$$w_S = w_0 + v_S \qquad w_T = w_0 + v_T$$

Encode them into SVMs:

Regularization terms for multiple tasks

$$\min_{w_0, v_t, \xi_{t_i}} \left\{ J(w_0, v_t, \xi_{t_i}) = \sum_{t\in\{S,T\}} \sum_{i=1}^{n_t} \xi_{t_i} + \frac{\lambda_1}{2} \sum_{t\in\{S,T\}} \|v_t\|^2 + \lambda_2 \|w_0\|^2 \right\}$$

$$s.t. \quad y_{t_i}(w_0 + v_t) \cdot x_{t_i} \geq 1 - \xi_{t_i},\ \xi_{t_i} \geq 0,\ i \in \{1, 2, ..., n_t\}\ and\ t \in \{S, T\}$$

Inductive Transfer Learning

Relational-knowledge-transfer Approaches

TAMAR[Mihalkova et al. AAAI-07]

Assumption: If the target domain and source domain are related, then there may be some relationship between domains being similar, which can be used for transfer learning

Input:

1. Relational data in the source domain and a statistical relational model, Markov Logic Network (MLN), which has been learnt in the source domain.
2. Relational data in the target domain.

Output: A new statistical relational model, MLN, in the target domain.

Goal: To learn a MLN in the target domain more efficiently and effectively.

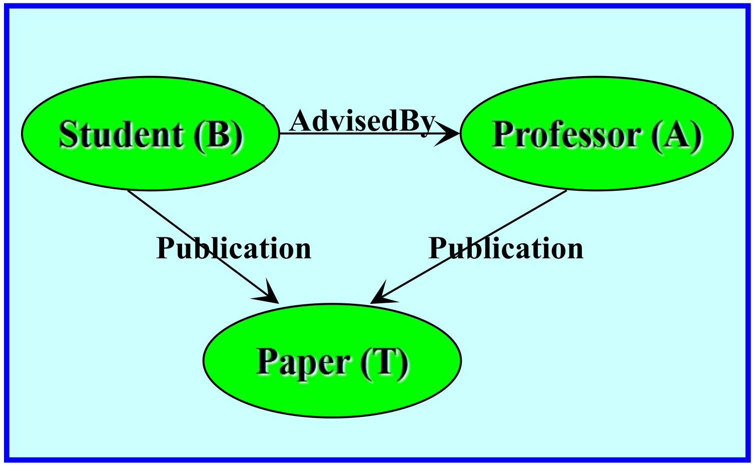
TAMAR [Mihalkova et al. AAAI-07]

Two Stages:

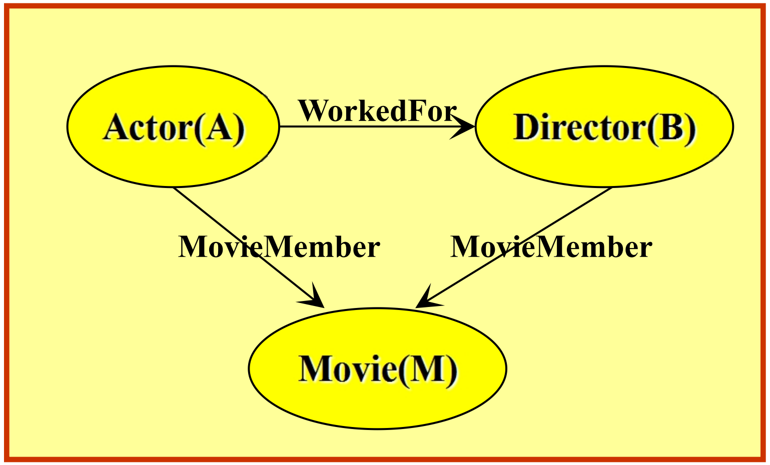
1. **Predicate Mapping**
   - Establish the mapping between predicates in the source and target domain. Once a mapping is established, clauses from the source domain can be translated into the target domain.
2. **Revising the Mapped Structure**
   - The clauses mapping from the source domain directly may not be completely accurate and may need to be revised, augmented , and re-weighted in order to properly model the target data.

TAMAR [Mihalkova et al. AAAI-07]

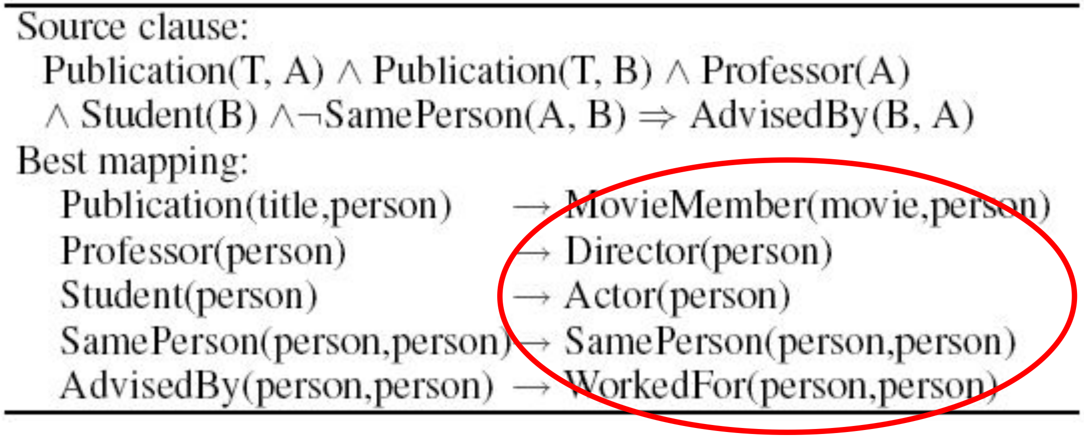
Source domain (academic domain)



Target domain (movie domain)



Mapping



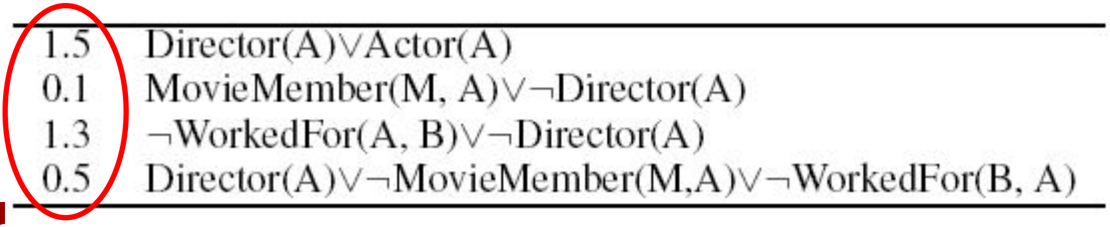
Source clause:

$\text{Publication}(T, A) \wedge \text{Publication}(T, B) \wedge \text{Professor}(A) \wedge \text{Student}(B) \wedge \neg \text{SamePerson}(A, B) \Rightarrow \text{AdvisedBy}(B, A)$

Best mapping:

Publication(title,person)	→	MovieMember(movie,person)
Professor(person)	→	Director(person)
Student(person)	→	Actor(person)
SamePerson(person,person)	→	SamePerson(person,person)
AdvisedBy(person,person)	→	WorkedFor(person,person)

Revising



1.5	$\text{Director}(A) \vee \text{Actor}(A)$
0.1	$\text{MovieMember}(M, A) \vee \neg \text{Director}(A)$
1.3	$\neg \text{WorkedFor}(A, B) \vee \neg \text{Director}(A)$
0.5	$\text{Director}(A) \vee \neg \text{MovieMember}(M, A) \vee \neg \text{WorkedFor}(B, A)$

Transductive Transfer Learning

Instance-transfer Approaches

Sample Selection Bias / Covariance Shift

[Zadrozny ICML-04, Schwaighofer JSPI-00]

Input: A lot of labeled data in the source domain and no labeled data in the target domain.

Output: Models for use in the target domain data.

Assumption: The source domain and target domain are the same. In addition, $P(Y_S \mid X_S)$ and $P(Y_T \mid X_T)$ are the same while $P(X_S)$ and $P(X_T)$ may be different causing by different sampling process (training data and test data).

Main Idea: Re-weighting (important sampling) the source domain data.

Sample Selection Bias/Covariance Shift

To correct sample selection bias:

$$\theta^* = \arg\min_{\theta \in \Theta} \sum_{i=1}^{n_S} \frac{\mathcal{P}(x_{T_i})}{\mathcal{P}(x_{S_i})} l(x_{S_i}, y_{S_i}, \theta)$$

weights for source domain data

How to estimate $\frac{\mathcal{P}(x_{T_i})}{\mathcal{P}(x_{S_i})}$ **?**

One straightforward solution is to estimate $P(X_S)$ and $P(X_T)$, respectively. However, estimating density function is a hard problem.

Sample Selection Bias/Covariance Shift

Kernel Mean Match (KMM)

[Huang et al. NIPS 2006]

Main Idea: KMM tries to estimate $\beta_i = \frac{\mathcal{P}(x_{S_i})}{\mathcal{P}(x_{T_i})}$ directly instead of estimating density function.

It can be proved that β_i can be estimated by solving the following quadratic programming (QP) optimization problem.

$$\min_{\beta} \quad \frac{1}{2}\beta^T K \beta - \kappa^T \beta$$

$$s.t. \quad \beta_i \in [0, B] \;\; and \;\; \left|\sum_{i=1}^{n_S} \beta_i - n_S\right| \leq n_S \epsilon$$

To match means between training and test data in a RKHS

Theoretical Support: Maximum Mean Discrepancy (MMD) [Borgwardt et al. BIOINFOMATICS-06]. The distance of distributions can be measured by Euclid distance of their mean vectors in a RKHS.

Transductive Transfer Learning

Feature-representation-transfer Approaches

Domain Adaptation

[Blitzer et al. EMNL-06, Ben-David et al. NIPS-07, Daume III ACL-07]

Assumption: Single task across domains, which means $P(Y_S \mid X_S)$ and $P(Y_T \mid X_T)$ are the same while $P(X_S)$ and $P(X_T)$ may be different causing by feature representations across domains.

Main Idea: Find a “good” feature representation that reduce the “distance” between domains.

Input: A lot of labeled data in the source domain and only unlabeled data in the target domain.

Output: A common representation between source domain data and target domain data and a model on the new representation for use in the target domain.

Domain Adaptation

Structural Correspondence Learning (SCL)

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

Motivation: If two domains are related to each other, then there may exist some "pivot" features across both domain. Pivot features are features that behave in the same way for discriminative learning in both domains.

Main Idea: To identify correspondences among features from different domains by modeling their correlations with pivot features. Non-pivot features form different domains that are correlated with many of the same pivot features are assumed to correspond, and they are treated similarly in a discriminative learner.

Unsupervised Transfer Learning

Feature-representation-transfer Approaches

Input: A lot of unlabeled data in a source domain and a few unlabeled data in a target domain.

Goal: Clustering the target domain data.

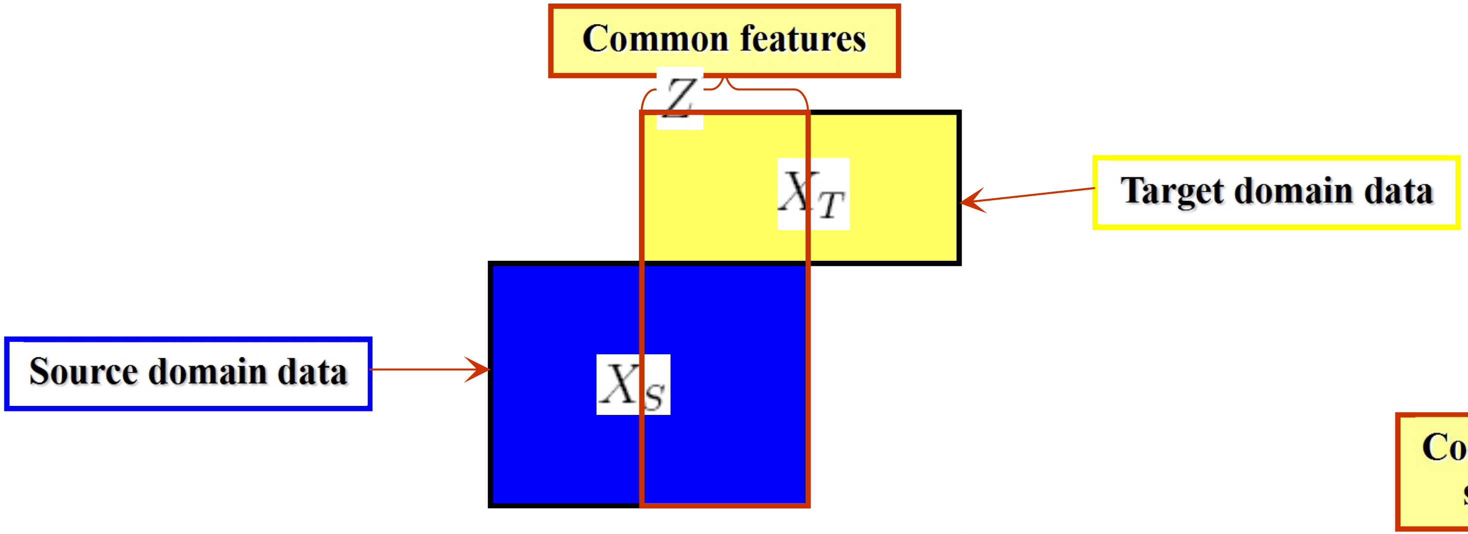
Assumption: The source domain and target domain data share some common features, which can help clustering in the target domain.

Main Idea: To extend the information theoretic co-clustering algorithm [Dhillon et al. KDD-03] for transfer learning.

- **Self-taught Clustering (STC)[Dai et al. ICML-08]**

Self-taught Clustering (STC)

[Dai et al. ICML-08]


Common features

Target domain data

Source domain data

Co-clustering in the source domain

Objective function that need to be minimized

$$J(\tilde{X}_T,\tilde{X}_S,\tilde{Z}) = \boxed{I(X_T,Z) - I(\tilde{X}_T,\tilde{Z})} + \lambda\boxed{\left[I(X_S,Z) - I(\tilde{X}_S,\tilde{Z})\right]}$$

Co-clustering in the target domain

where $\boxed{C_{X_T}} : X_T \to \tilde{X}_T, \boxed{C_{X_S}} : X_S \to \tilde{X}_S$ and $\boxed{C_Z} : Z \to \tilde{Z}$

Output

Cluster functions

Negative Transfer

- Most approaches to transfer learning assume transferring knowledge across domains be always positive.
- However, in some cases, when two tasks are too dissimilar, brute-force transfer may even hurt the performance of the target task, which is called negative transfer [Rosenstein et al NIPS-05 Workshop].
- Some researchers have studied how to measure relatedness among tasks [Ben-David and Schuller NIPS-03, Bakker and Heskes JMLR-03].
- How to design a mechanism to avoid negative transfer needs to be studied theoretically.

Conclusion

	Inductive Transfer Learning	Transductive Transfer Learning	Unsupervised Transfer Learning
Instance-transfer	√	√	
Feature-representation-transfer	√	√	√
Model-transfer	√		
Relational-knowledge-transfer	√		

How to avoid negative transfer need to be attracted more attention!

THANKS

Q & A