An Introduction to Transfer Learning (迁移学习)

Junhao Hua

2015/1/16

Supervised learning

Train | Test

Supervised learning algorithms may not work well with limited labeled data.

What is the transfer learning ?

- Traditional Machine Learning Algorithm
  - Make predictions using previously collected labeled or unlabeled training data.
  - Semisupervised: built a good classifier using a large amount of unlabeled data and a small amount of labeled data.
- Transfer Learning
  - 1995 NIPS: “Learning to Learn”, life-long learning, knowledge transfer, inductive transfer, multitask learning, metalearning, etc.
  - 2005 new Mission: <u>the ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks</u>.

What is the transfer learning ?

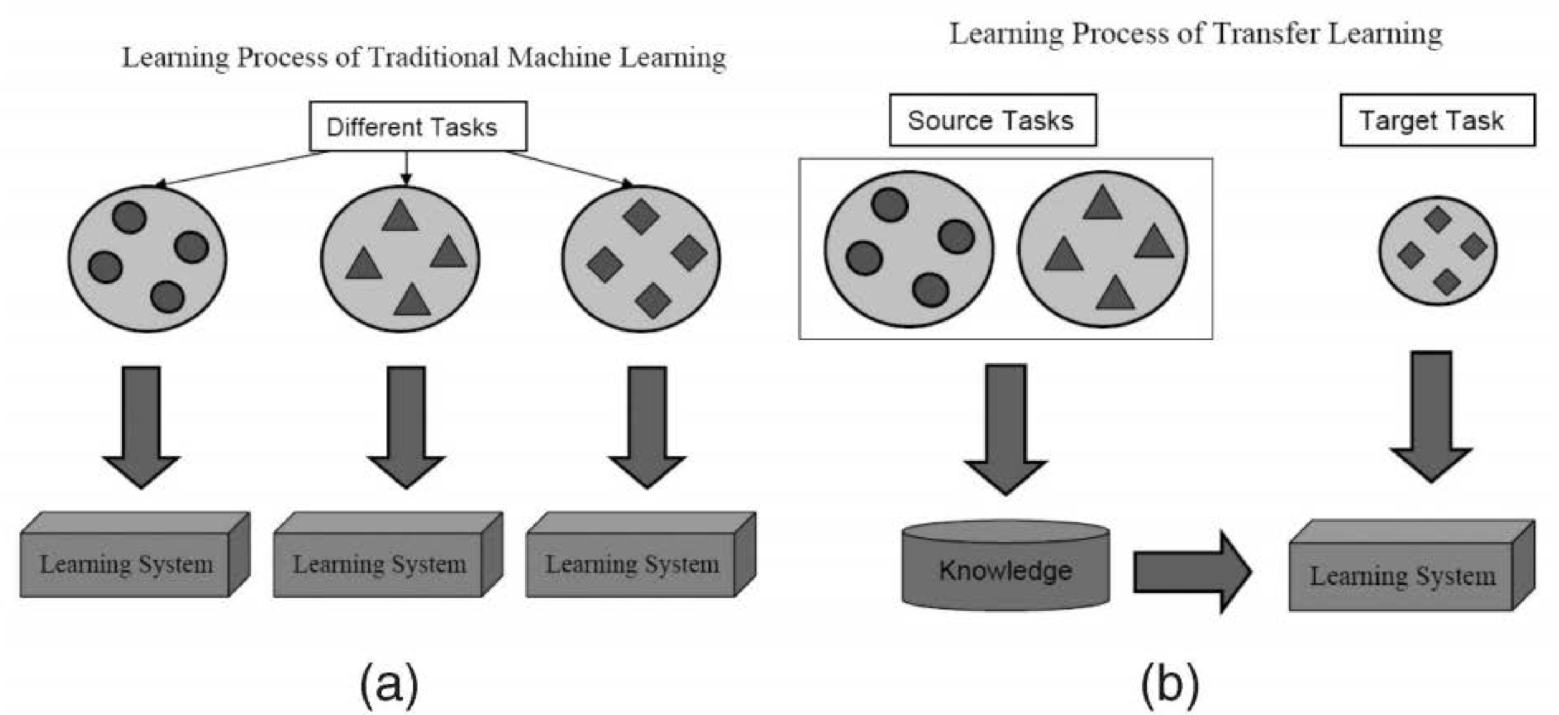


Fig. 1. Different learning processes between (a) traditional machine learning and (b) transfer learning.

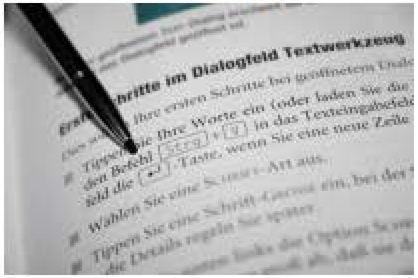
Domain: D = {x, P(x)}
Tasks: T = {Y, f(.)=P(Y|X)}
Labeled or unlabeled?

Transfer Learning: D_s~=D_t || T_s ~= T_t

- The <u>domains</u> {X, P(X)} are different:
  - The feature spaces X_s ~= X_t
    - Eg., use different languages in document classification example;
  - or marginal distributions, P_s(X) ~= P_t(X)
    - Eg., focus on different topics.
- The <u>tasks</u> {Y, P(Y|X)} are different:
  - The label spaces Y_s ~= Y_t
    - Eg., the source domain has binary document classes whereas the target domain has 10 classes.
  - or Cond. prob. Distr., P(Y_s|X_s) ~= P(Y_t|X_t)
    - Eg., the source and target documents are very unbalanced.

Sentiments from Amazon

- Target task
  - Kitchen appliances
- Source domains(Described with same language)
  - Book DVDs Electronics



- Can we leverage existing labeled data from other source domains ?

NANYANG TECHNOLOGICAL UNIVERSITY

Photos from Different Domains

high quality

low quality

daylight

sunset

posed

“in the wild”

art

surveillance

Examples are taken from Saenko

NANYANG TECHNOLOGICAL UNIVERSITY

TABLE 1
Relationship between Traditional Machine Learning and Various Transfer Learning Settings

Learning Settings		Source and Target Domains	Source and Target Tasks
Traditional Machine Learning		the same	the same
Transfer Learning	*Inductive Transfer Learning /*	the same	different but related
	Unsupervised Transfer Learning	different but related	different but related
	Transductive Transfer Learning	different but related	the same

Domains D = {X, P(X)}
- Source(D_s), Target(D_t), 0<n_t << n_s

Tasks T = {Y, f(.)=P(Y|X)}:
- Source(T_s), Target(T_t)

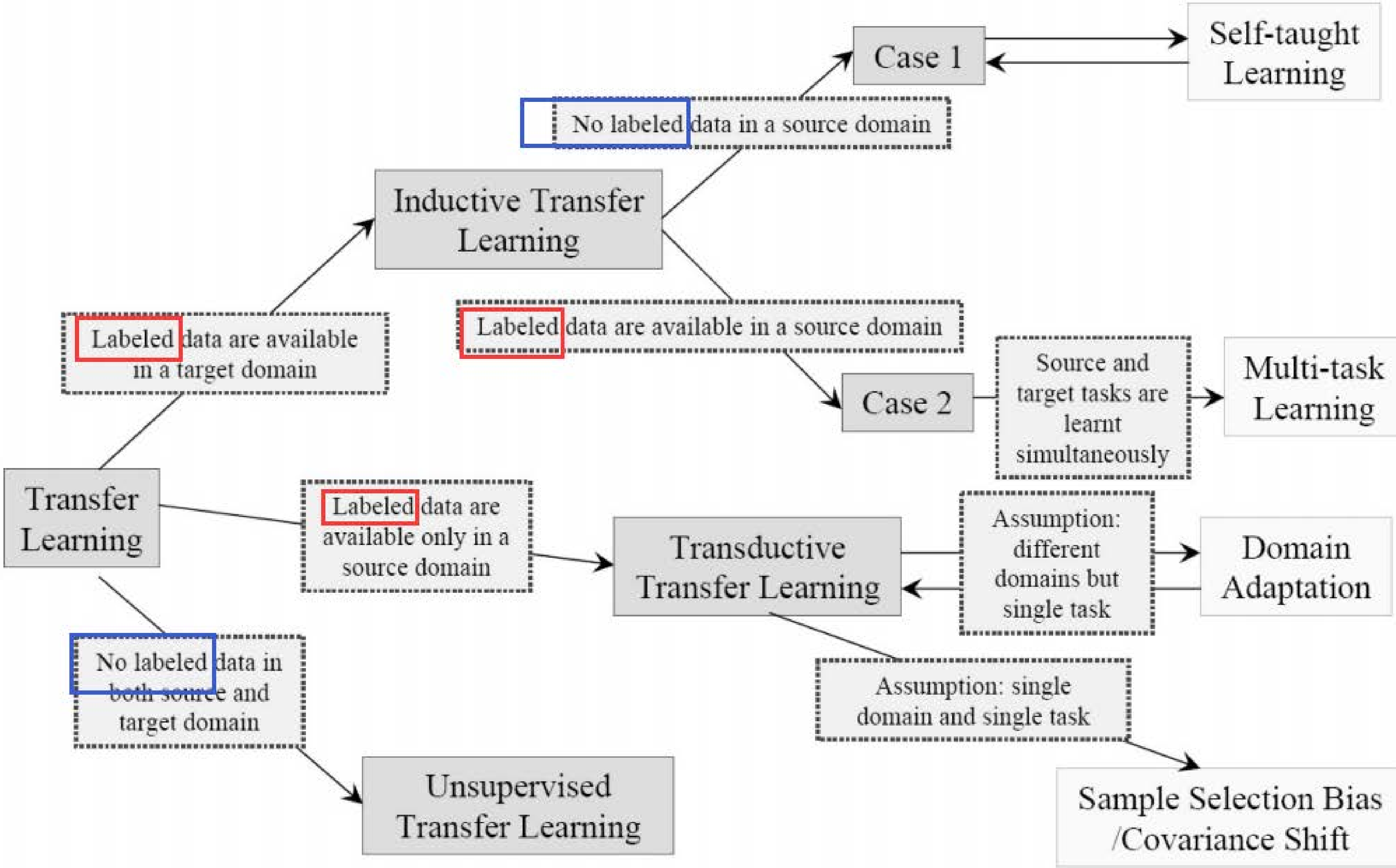
D_s ~=D_t or T_s ~= T_s

related ?

TABLE 2
Different Settings of Transfer Learning

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
Inductive Transfer Learning	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
Unsupervised Transfer Learning		Unavailable	Unavailable	Clustering, Dimensionality Reduction

Only a few or even no labels



. 2. An overview of different settings of transfer.

Kinds of transfer learning

- Inductive(归纳式) Transfer Learning
  - Multi-task Learning
  - Self-taught Learning
- Transductive(直推式) Transfer Learning
  - Domain Adaptation
  - Sample Selection Bias, Co-variate Shift
- Unsupervised Transfer Learning

Multi-task Learning

- Learning multiple related tasks simultaneously.
- Motivation
  - Only a few data per task available;
  - Shown to improve performance relative to learning each task independently.
- Task relatedness
  - Modeled by assuming all functions learned are close to each other in some norm [Bakker03, Evgeniou05];
  - Share some parameters or prior (GP) distributions of hyperparameters; Hierarchical Bayes with GP, etc. [Lawrence04, Bonilla08, Evgeniou04];
  - Share a common underlying representation [David03, Evgeniou07];

Multi-task Reference

- Evgeniou, Theodoros, Charles A. Micchelli, and Massimiliano Pontil. "Learning multiple tasks with kernel methods." *Journal of Machine Learning Research*. 2005.
- Evgeniou, A., and Massimiliano Pontil. "Multi-task feature learning." *Advances in neural information processing systems* 19 (2007): 41.
- Argyriou, Andreas, Theodoros Evgeniou, and Massimiliano Pontil. "Convex multi-task feature learning." *Machine Learning* 73.3 (2008): 243-272.
- Ben-David, Shai, and Reba Schuller. "Exploiting task relatedness for multiple task learning." *Learning Theory and Kernel Machines*. Springer Berlin Heidelberg, 2003. 567-580.
- Yu, Kai, Volker Tresp, and Anton Schwaighofer. "Learning Gaussian processes from multiple tasks." *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005.
- Lawrence, Neil D., and John C. Platt. "Learning to learn with the informative vector machine." *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.
- Bonilla, Edwin, Kian Ming Chai, and Christopher Williams. "Multi-task Gaussian process prediction." (2008).
- Evgeniou, Theodoros, and Massimiliano Pontil. "Regularized multi--task learning." *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004.

Example: Multi-Task Feature Learning [Evgeniou07]

- The Basic Idea: to learn a low-dimensional representation that is shared across related tasks.
- Common features can be learned by

linear features

$$\arg\min_{A,U} \sum_{t\in\{T,S\}} \sum_{i=1}^{n_t} L(y_{t_i}, \langle a_t, \boxed{U^T x_{t_i}} \rangle) + \gamma \|A\|_{2,1}^2 \quad (1)$$

$$s.t. \quad U \in \mathbf{O}^d.$$

  - U is a d x d orthogonal matrix (mapping function);
  - A = [a_s, a_t] is a matrix of parameters;
- Then it was transformed into an equivalent convex problem, following an alternately procedure.

Self-taught learning [Raina07]

- Motivation
  - Labeled data is expensive to obtain.
  - It's difficult even to obtain many unlabeled examples in the target domain.
  - How can use unlabeled data (images, etc.) from other object classes which are much easier to obtain ?
- Technique
  - Uses sparse coding to construct higher-level features using the unlabeled data;
  - Apply this representation to the target data;
  - Then use it for the classification task.

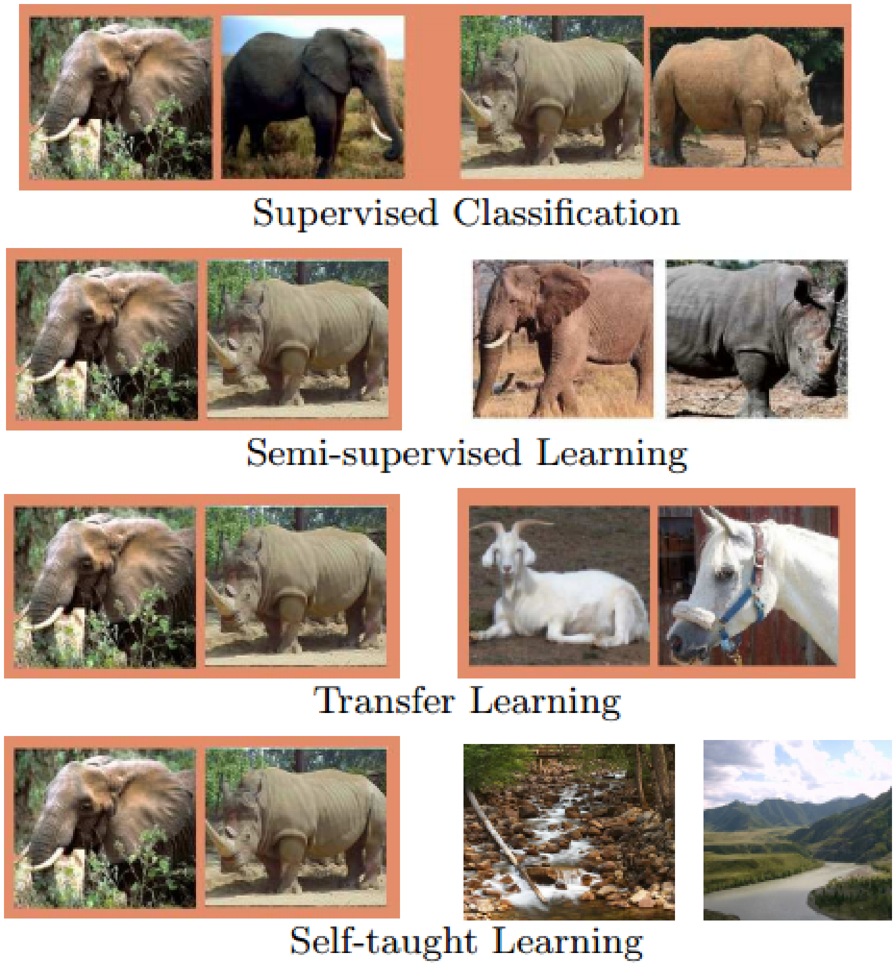
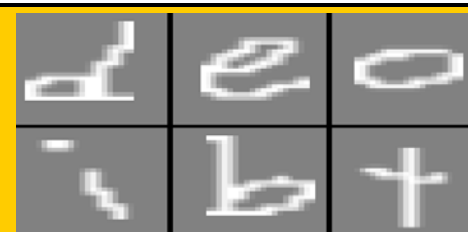


Figure 1. Machine learning formalisms for classifying images of elephants and rhinos. Images on orange background are labeled; others are unlabeled. Top to bottom: Supervised classification uses labeled examples of elephants and rhinos; semi-supervised learning uses additional unlabeled examples of elephants and rhinos; transfer learning uses additional labeled datasets; self-taught learning just requires additional unlabeled images, such as ones randomly downloaded from the Internet.

“Self-taught Learning”

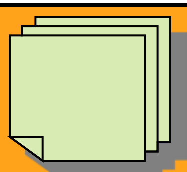
Labeled Digits

+



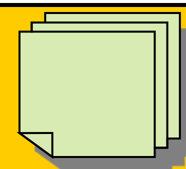
Unlabeled English characters

?



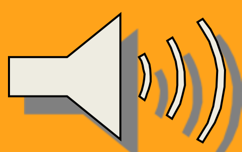
Labeled Webpages

+



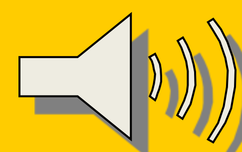
Unlabeled newspaper articles

?



Labeled Russian Speech

+



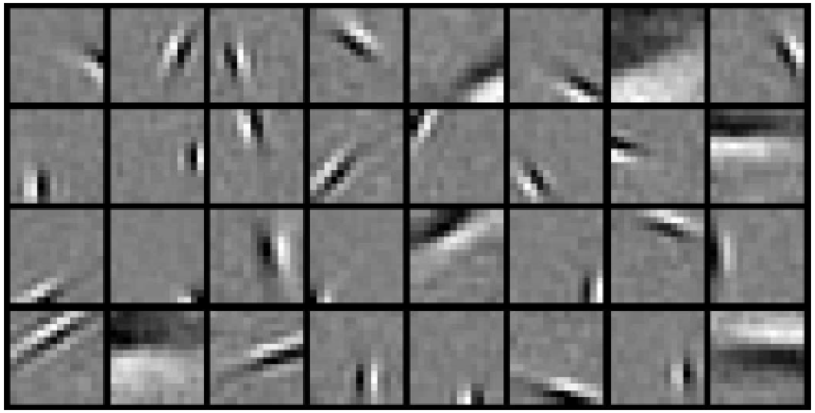
Unlabeled English speech

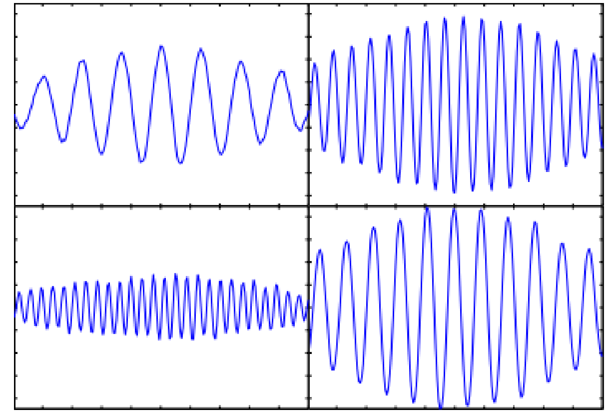
?

Self-taught learning

$$\begin{aligned} \text{minimize}_{b,a} \quad & \textstyle\sum_i \|x_u^{(i)} - \sum_j \underbrace{\boxed{a_j^{(i)}}}_{\text{Sparse features}} \underbrace{\boxed{b_j}}_{\text{Basis vector}} \|_2^2 + \beta \|a^{(i)}\|_1 \quad (1) \\ \text{s.t.} \quad & \|b_j\|_2 \leq 1, \quad \forall j \in 1, ..., s \end{aligned}$$

- Notes:
  - Base size s >> dimension n;
  - Encourage the activations(features) a to be sparse;
  - Features a(x) are inherently nonlinear function.
- Problem is convex over variable a (b);
- Iteratively optimized over a and b alternatingly.
- Application: Deep Neural Networks (Deep Learning)





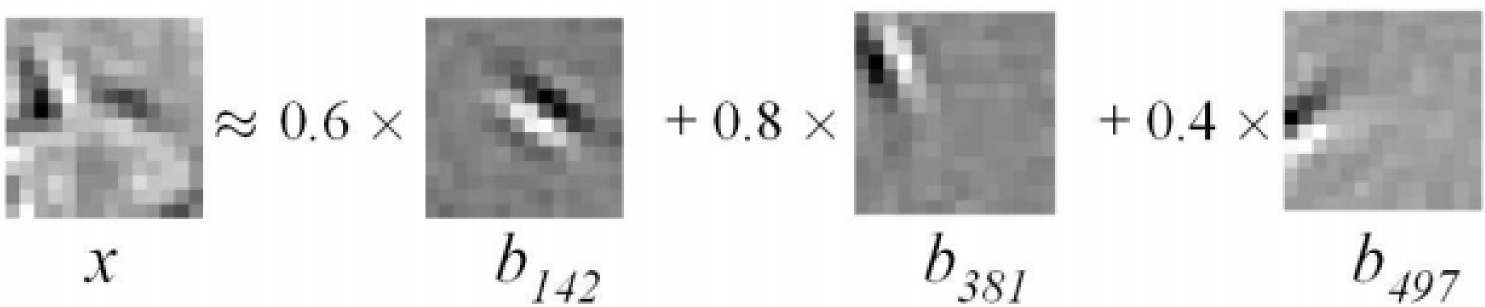


Figure 3. The features computed for an image patch (left) by representing the patch as a sparse weighted combination of bases (right). These features act as robust edge detectors.

Self-taught learning

Algorithm 1 Self-taught Learning via Sparse Coding

input Labeled training set
$T = \{(x_l^{(1)}, y^{(1)}), (x_l^{(2)}, y^{(2)}), \ldots, (x_l^{(m)}, y^{(m)})\}$.
Unlabeled data $\{x_u^{(1)}, x_u^{(2)}, \ldots, x_u^{(k)}\}$.

output Learned classifier for the classification task.

algorithm Using unlabeled data $\{x_u^{(i)}\}$, solve the optimization problem (1) to obtain bases b.
Compute features for the classification task to obtain a new labeled training set $\hat{T} = \{(\hat{a}(x_l^{(i)}), y^{(i)})\}_{i=1}^{m}$, where
$\hat{a}(x_l^{(i)}) = \arg\min_{a^{(i)}} \|x_l^{(i)} - \sum_j a_j^{(i)} b_j\|_2^2 + \beta \|a^{(i)}\|_1$.
Learn a classifier $\mathcal{C}$ by applying a supervised learning algorithm (e.g., SVM) to the labeled training set $\hat{T}$.
return the learned classifier $\mathcal{C}$.

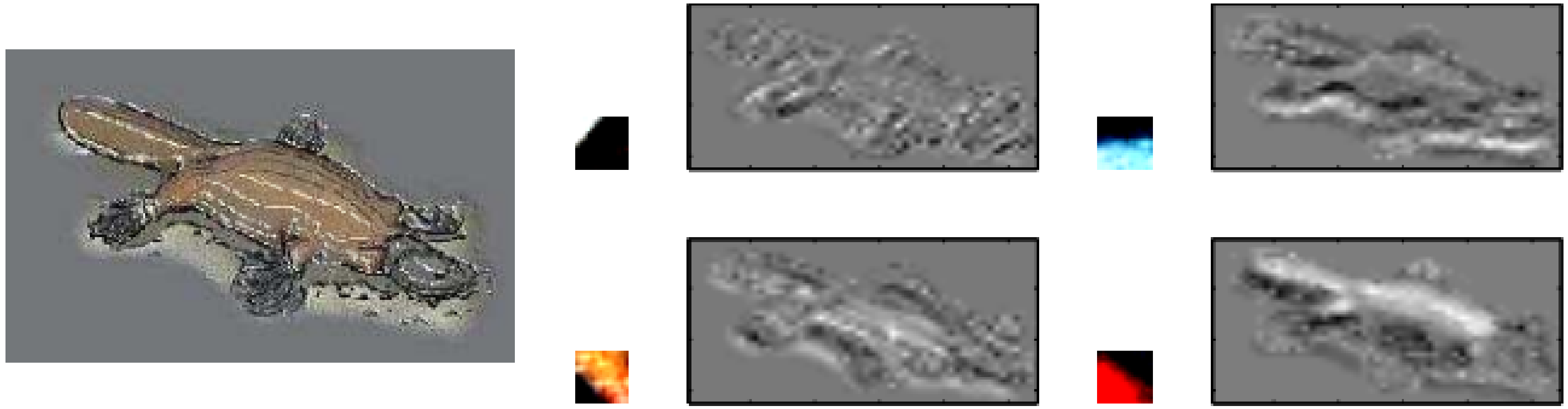


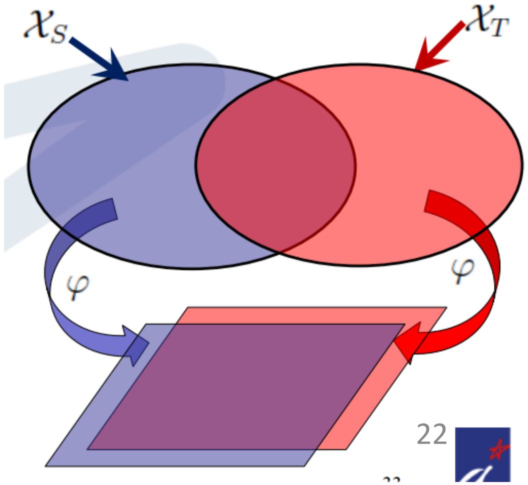
Figure 4. Left: An example platypus image from the Caltech 101 dataset. Right: Features computed for the platypus image using four sample image patch bases (trained on color images, and shown in the small colored squares) by computing features at different locations in the image. In the large figures on the right, white pixels represents highly positive feature values for the corresponding basis, and black pixels represents highly negative feature values. These activations capture higher-level structure of the input image. (Bases have been magnified for clarity; best viewed in color.)

Reference for Self-taught learnig

- Raina, Rajat, et al. "Self-taught learning: transfer learning from unlabeled data."*Proceedings of the 24th international conference on Machine learning*. ACM, 2007.

Domain Adaptation

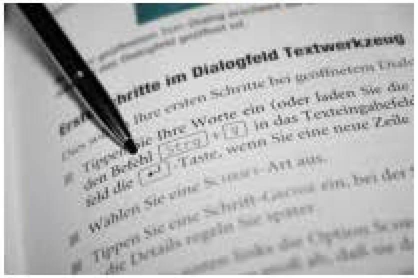
- Setting
  - The source and target tasks are the same.
  - The domains are different but related.
    - D_s ~= D_t;
    - The feature spaces are different, X_s ~= X_t;
  - A lot of labeled data in the source domain;
  - Only a few or even no labeled data in the target domain.
- Notes
  - Transfer Learning in NLP is referred
  - as domain Adaptation.



Sentiments from Amazon

- Target task
  - Kitchen appliances

- Source domains(Described with same language)
  - Book　　　　DVDs　　　　Electronics



- Can we leverage existing labeled data from other source domains ?

NANYANG TECHNOLOGICAL UNIVERSITY

Algorithms for Domain Adaptation

Structural correspondence learning [Blitzer06]

- Extract some relevant features
  - Treats M Pivot features (Determiners) as a new label vector.
  - Solve M Pivot predictors
  - L(.): the modified Huber loss.
  - SVD is applied to W

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

Output: predictor $f : X \to Y$

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

2. For $\ell = 1$ to m

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

end

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

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

Figure 3: SCL Algorithm

- The learned mapping \theta encodes the correspondence between the features from different domains.
- \theta x is the desired mapping to the (low dimensional) shared feature representation.

Domain adaptation problems: A DASVM classification technique and a circular validation strategy [Bruzzone10]

- Phase 1: initialization
  - Standard supervised SVMs

$$\begin{cases} \min_{\mathbf{w},b,\xi} \left\{ \frac{1}{2} \parallel \mathbf{w}^{(0)} \parallel^2 + C \sum_l \xi_l^s \right\} \\ y_l^s \left(\mathbf{w}^{(0)} \cdot \mathbf{x}_l^s + b^{(0)} \right) \geq 1 - \xi_l^s \quad \forall l = 1, \ldots, N, \ \left(\mathbf{x}_l^s, y_l^s \right) \in \mathcal{T}^{(0)}, \\ \xi_l^s \geq 0 \end{cases}$$

  - The separation hyperplane

$$h^{(0)} : \mathbf{w}^{(0)} \cdot \mathbf{x} + b^{(0)} = 0,$$

- Phase 2: Iterative Domain Adaptation
  - A subset of the (remaining) unlabeled samples x_t is iteratively selected and moved into the training set.

$$
\begin{cases}
\min_{\mathbf{w},b,\xi^s,\xi^t} \left\{ \frac{1}{2} \left\| \mathbf{w}^{(i)} \right\|^2 + C^{(i)} \sum_l \xi_l^s + \sum_u C_u^* \xi_u^t \right\} \\
y_l^s \cdot \left(\mathbf{w}^{(i)} \cdot \mathbf{x}_l^s + b^{(i)} \right) \geq 1 - \xi_l^s \\
\qquad \forall l = 1, \ldots, \mu^{(i)}, \left(\mathbf{x}_l^s, y_l^s \right) \in \mathcal{T}^{(i)} \\
\hat{y}_u^{t(i-1)} \cdot \left(\mathbf{w}^{(i)} \cdot \mathbf{x}_u^t + b^{(i)} \right) \geq 1 - \xi_u^t \\
\qquad \forall u = 1, \ldots, \eta^{(i)}, \left(\mathbf{x}_u^t, \hat{y}_u^{t(i-1)} \right) \in \mathcal{T}^{(i)} \\
\xi_l^s, \xi_u^t \geq 0.
\end{cases}
$$

- Phase 3: Convergence

empirical stopping criterion has been defined:

$$
\begin{cases}
\mathcal{Q}^{(i)} = \varnothing, \\
|\mathcal{H}^{(i)}| \leq \lceil \beta \cdot M \rceil, \\
|\mathcal{S}^{(i)}| \leq \lceil \beta \cdot M \rceil
\end{cases}
\tag{11}
$$

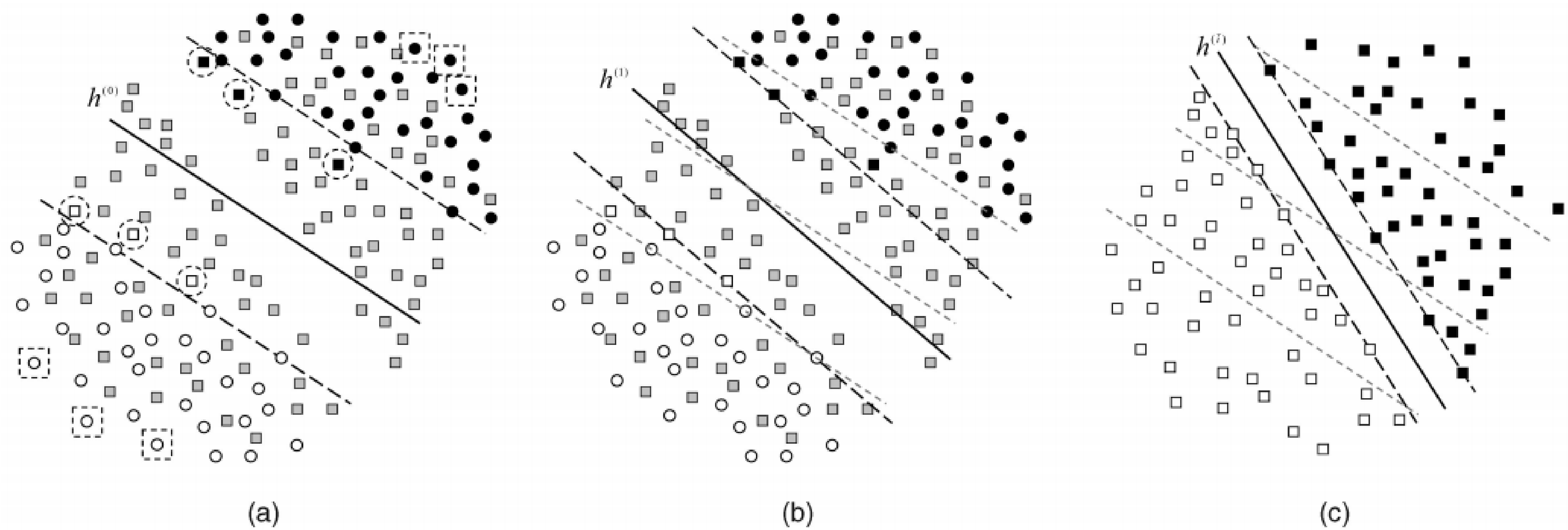


Fig. 1. Separation hyperplane (solid line) and margin bounds (dashed lines) at different stages of the DASVM algorithm for a toy data set. Labeled source-domain patterns are shown as white and black circles. Semilabeled target-domain patterns are shown as white and black squares, respectively. Unlabeled target-domain patterns are represented as gray squares. Feature space structure obtained: (a) at the first iteration (the dashed circles highlight the ρ semilabeled patterns selected from both sides of the margin; in the example $\rho = 3$); (b) at the second iteration and (c) at the last iteration, respectively, in an ideal situation (the dashed gray lines represent both the separation hyperplane and the margin bounds at the beginning of the learning process).

ADSVM: Circular validation strategy

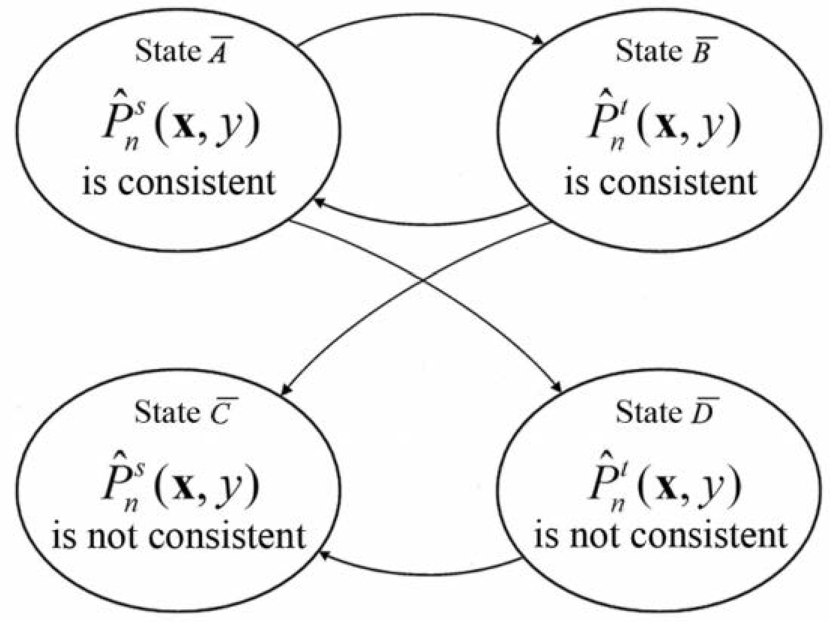


Fig. 3. Diagram of all the possible state transitions exploited from the proposed circular validation strategy.

$$\mathcal{A} = \{g_n(\mathbf{x}) \mid \Lambda(\mathcal{Y}_s, \hat{\mathcal{Y}}_{sn}) \geq \Lambda_{\text{th}}\},$$

$$\mathcal{B} = \{g_n(\mathbf{x}) \mid \Lambda(\mathcal{Y}_t, \hat{\mathcal{Y}}_{tn}) \geq \Lambda_{\text{th}}\},$$

$$\mathcal{C} = \{g_n(\mathbf{x}) \mid \Lambda(\mathcal{Y}_s, \hat{\mathcal{Y}}_{sn}) < \Lambda_{\text{th}}\},$$

$$\mathcal{D} = \{g_n(\mathbf{x}) \mid \Lambda(\mathcal{Y}_t, \hat{\mathcal{Y}}_{tn}) < \Lambda_{\text{th}}\},$$

If $g_n(\mathbf{x}) \in \mathcal{A}$, we assume that $\hat{P}_n^s(\mathbf{x}, y)$ is consistent with $P^s(\mathbf{x}, y)$ (the system is in state $\bar{A}$).
If $g_n(\mathbf{x}) \in \mathcal{B}$, we assume that $\hat{P}_n^t(\mathbf{x}, y)$ is consistent with $P^t(\mathbf{x}, y)$ (the system is in state $\bar{B}$).
If $g_n(\mathbf{x}) \in \mathcal{C}$, we assume that $\hat{P}_n^s(\mathbf{x}, y)$ is not consistent with $P^s(\mathbf{x}, y)$ (the system is in state $\bar{C}$).
If $g_n(\mathbf{x}) \in \mathcal{D}$, we assume that $\hat{P}_n^t(\mathbf{x}, y)$ is not consistent with $P^t(\mathbf{x}, y)$ (the system is in state $\bar{D}$).

Starting from state D, the system must never move back to state A
Starting from state B, the system can return to state A.

Domain Transfer Multiple Kernel Learning(DTMKL) [Duan12]

- Setting
  - Labeled source data & a limited number of labeled target data.
- Frameworks
  - To learn decision function and kernel function simultaneously.

$$f(\mathbf{x}) = \mathbf{w}'\phi(\mathbf{x}) + b = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b,$$

  - Reducing Mismatch of Data Distribution & the structural risk functional of any kernel method (SVM, SVR, KRLS, etc.).

$$[k, f] = \arg\min_{k,f} \ \Omega\left(\boxed{\mathrm{DIST}_k^2}(D^A, D^T)\right) + \theta \boxed{R(k, f, D)},$$

    - \Omega is any monotonic increasing function
    - R(.) is defined on the labeled patterns.

DTMKL [Duan12]

- Maximum Mean Discrepancy (MMD) [Borgwardt06]

$$\begin{aligned}\mathrm{DIST}_k(D^A, D^T) &= \sup_{\|f\|_{\mathcal{H}}\le 1} \left(\mathrm{E}_{\mathbf{x}^A\sim\mathcal{Q}}[f(\mathbf{x}^A)] - \mathrm{E}_{\mathbf{x}^T\sim\mathcal{P}}[f(\mathbf{x}^T)]\right) \\ &= \sup_{\|f\|_{\mathcal{H}}\le 1} \left\langle f, \left(\mathrm{E}_{\mathbf{x}^A\sim\mathcal{Q}}[\phi(\mathbf{x}^A)] - \mathrm{E}_{\mathbf{x}^T\sim\mathcal{P}}[\phi(\mathbf{x}^T)]\right)\right\rangle_{\mathcal{H}} \\ &= \left\|\mathrm{E}_{\mathbf{x}^A\sim\mathcal{Q}}[\phi(\mathbf{x}^A)] - \mathrm{E}_{\mathbf{x}^T\sim\mathcal{P}}[\phi(\mathbf{x}^T)]\right\|_{\mathcal{H}},\end{aligned}$$

  - Can be estimated by

$$\mathrm{DIST}_k(D^A, D^T) = \left\|\frac{1}{n_A}\sum_{i=1}^{n_A}\phi(\mathbf{x}_i^A) - \frac{1}{n_T}\sum_{i=1}^{n_T}\phi(\mathbf{x}_i^T)\right\|_{\mathcal{H}}.$$

$$\mathrm{DIST}_k^2(D^A, D^T) = \|\Phi\mathbf{s}\|^2 = \mathrm{tr}(\Phi'\Phi\mathbf{S}) = \mathrm{tr}(\mathbf{KS}),$$

where

$$\mathbf{S} = \mathbf{s}\mathbf{s}' \in \Re^{(n_A+n_T)\times(n_A+n_T)}, \quad \mathbf{K} = \Phi'\Phi = \begin{bmatrix} \mathbf{K}^{A,A} & \mathbf{K}^{A,T} \\ \mathbf{K}^{T,A} & \mathbf{K}^{T,T} \end{bmatrix}$$

$$\in \Re^{(n_A+n_T)\times(n_A+n_T)}, \quad \mathbf{K}^{A,A} \in \Re^{n_A\times n_A}, \quad \mathbf{K}^{T,T} \in \Re^{n_T\times n_T},$$

DTMKL [Duan12]

- Multiple Base Kernels [Lanckriet04]
  - Assume kernel k is a linear combination of base kernels:

$$k = \sum_{m=1}^{M} d_m k_m,$$

  - Then

$$\Omega(\mathrm{tr}(\mathbf{KS})) = \frac{1}{2}(\mathrm{tr}(\mathbf{KS}))^2$$
$$= \frac{1}{2}\left(\mathrm{tr}\left(\sum_{m=1}^{M} d_m \mathbf{K}_m \mathbf{S}\right)\right)^2 = \frac{1}{2}\mathbf{d}'\mathbf{p}\mathbf{p}'\mathbf{d},$$

where

$$\mathbf{p} = [p_1, \ldots, p_M]', p_m = \mathrm{tr}(\mathbf{K}_m\mathbf{S}), \mathbf{K}_m = [k_m(\mathbf{x}_i, \mathbf{x}_j)] \in \Re^{(n_A+n_T)\times(n_A+n_T)},$$

and $\mathbf{d} = [d_1, \ldots, d_M]'$. Moreover, from (3), we have $f(\mathbf{x}) = \sum_{m=1}^{M} d_m \mathbf{w}_m' \phi_m(\mathbf{x}) + b$, where $\mathbf{w}_m = \sum_{i=1}^{n} \alpha_i \phi_m(\mathbf{x}_i)$.

Thus, the optimization problem in (4) can be rewritten as

$$\boxed{\min_{\mathbf{d}\in\mathcal{D}} \min_{f} \frac{1}{2}\mathbf{d}'\mathbf{p}\mathbf{p}'\mathbf{d} + \theta\, R(\mathbf{d}, f, D),} \qquad (5)$$

DTMKL [Duan12]

Algorithm 1. DTMKL Algorithm.

1: Initialize $\mathbf{d} = \frac{1}{M}\mathbf{1}_M$.

2: For $t = 1, \ldots, T_{\max}$

3: Solve the target classifer f in the objective function in (6).

$$J(\mathbf{d}) = \min_f \boxed{R(\mathbf{d}, f, D)}. \tag{6}$$

4: Update the linear combination coefficient vector $\mathbf{d}$ of multiple base kernels using (8).

$$\boxed{\mathbf{d}_{t+1} = \mathbf{d}_t - \eta_t \mathbf{g}_t \in \mathcal{D},} \tag{8}$$

5: End.

Using Hinge Loss(SVM)
Using Existing Base Classifiers

$$\boxed{\min_{\mathbf{d} \in \mathcal{D}} \; h(\mathbf{d}) = \min_{\mathbf{d} \in \mathcal{D}} \; \frac{1}{2}\mathbf{d}'\mathbf{p}\mathbf{p}'\mathbf{d} + \theta \, J(\mathbf{d}).}$$

Reference for Domain Adaptation

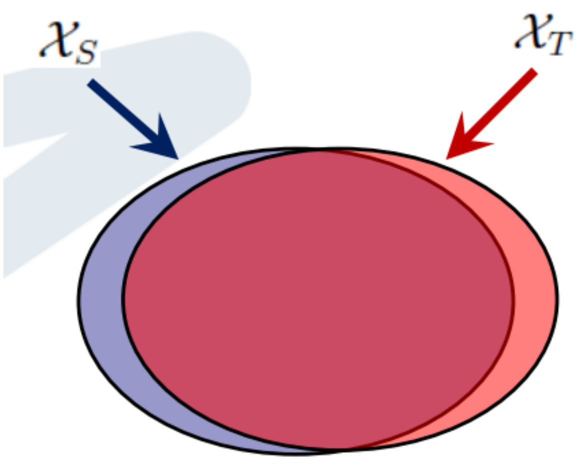
- Bruzzone, Lorenzo, and Mattia Marconcini. "Domain adaptation problems: A DASVM classification technique and a circular validation strategy." Pattern Analysis and Machine Intelligence, IEEE Transactions on 32.5 (2010): 770-787.
- Xing, Dikan, et al. "Bridged refinement for transfer learning." Knowledge Discovery in Databases: PKDD 2007. Springer Berlin Heidelberg, 2007. 324-335.
- Blitzer, John, Ryan McDonald, and Fernando Pereira. "Domain adaptation with structural correspondence learning." Proceedings of the 2006 conference on empirical methods in natural language processing. Association for Computational Linguistics, 2006.
- Hal Daume, III, Abhishek Kumar, and Avishek Saha. 2010. Frustratingly easy semi-supervised domain adaptation. In Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing (DANLP 2010). Association for Computational Linguistics, Stroudsburg, PA, USA, 53-59.
- Duan, Lixin, Ivor W. Tsang, and Dong Xu. "Domain transfer multiple kernel learning." Pattern Analysis and Machine Intelligence, IEEE Transactions on34.3 (2012): 465-479.
- Ando, Rie Kubota, and Tong Zhang. "A framework for learning predictive structures from multiple tasks and unlabeled data." The Journal of Machine Learning Research 6 (2005): 1817-1853.

Reference for Domain Adaptation

- Yang, Jun, Rong Yan, and Alexander G. Hauptmann. "Cross-domain video concept detection using adaptive svms." Proceedings of the 15th international conference on Multimedia. ACM, 2007.
- Borgwardt, Karsten M., et al. "Integrating structured biological data by kernel maximum mean discrepancy." *Bioinformatics* 22.14 (2006): e49-e57.
- Lanckriet, Gert RG, et al. "Learning the kernel matrix with semidefinite programming." *The Journal of Machine Learning Research* 5 (2004): 27-72.

Sample Selection Bias/Co-variate Shift

- The same setting Domain adaptation but
  - D_s ~= D_t
  - The feature spaces are the same, X_s = x_T (set)
  - The marginal prob. Distr. are different: P(X_s) ~= P(X_t)
  - if P(y_s|x_s) ~= P(y_t|x_t), then sample selection bias;
  - If P(y_s|x_s) \approx P(y_t|x_t) then covariate shift.



Sample Selection Bias/Co-variate Shift

- Technique: sample reweight

$$\theta^* = \arg\min_{\theta \in \Theta} \mathbb{E}_{(x,y)\in P}[l(x, y, \theta)],$$

$$\downarrow$$

$$\theta^* = \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} [l(x_i, y_i, \theta)],$$

$$\downarrow$$

$$\theta^* = \arg\min_{\theta \in \Theta} \sum_{(x,y)\in D_S} P(D_S) l(x, y, \theta).$$

$$\downarrow$$

$$\theta^* = \arg\min_{\theta \in \Theta} \sum_{(x,y)\in D_S} \frac{P(D_T)}{P(D_S)} P(D_S) l(x, y, \theta)$$

$$\approx \arg\min_{\theta \in \Theta} \sum_{i=1}^{n_S} \boxed{\frac{P_T(x_{T_i}, y_{T_i})}{P_S(x_{S_i}, y_{S_i})}} l(x_{S_i}, y_{S_i}, \theta).$$

- Estimate Independently[Zadrozny04]
- Kernel-mean matching[Huang07]
- KL importance estimation[Sugiyama08]

$$P(Y_T|X_T) = P(Y_S|X_S).$$

$$\boxed{\frac{P_T(x_{T_i}, y_{T_i})}{P_S(x_{S_i}, y_{S_i})} = \frac{P(x_{S_i})}{P(x_{T_i})}}.$$

Reference for Sample Selection Bias

- Zadrozny, Bianca. "Learning and evaluating classifiers under sample selection bias." *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.
- Huang, Jiayuan, et al. "Correcting sample selection bias by unlabeled data."*Advances in neural information processing systems*. 2006.
- Shimodaira, Hidetoshi. "Improving predictive inference under covariate shift by weighting the log-likelihood function." *Journal of statistical planning and inference* 90.2 (2000): 227-244.

- Huang, Jiayuan, et al. "Correcting sample selection bias by unlabeled data."*Advances in neural information processing systems*. 2006.

- Sugiyama, Masashi, et al. "Direct importance estimation with model selection and its application to covariate shift adaptation." *Advances in neural information processing systems*. 2008.

Unsupervised Transfer Learning

- Clustering, Dimensionality Reduction
  - Self-taught clustering[Dai08], minimize,

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

    - Z is a shared feature space by X_s and X_t
    - I(.,.) is the mutual information.

- Transferred discriminative analysis [Wang08], iteratively,
  - applies clustering methods to generate pseudoclass labels for the target unlabeled data.
  - applies dimensionality reduction methods to the target data and labeled source data to reduce the dimensions.

Reference for Unsupervised Transfer Learning

- Dai, Wenyuan, et al. "Self-taught clustering." *Proceedings of the 25th international conference on Machine learning*. ACM, 2008.
- Wang, Zheng, Yangqiu Song, and Changshui Zhang. "Transferred dimensionality reduction." *Machine learning and knowledge discovery in databases*. Springer Berlin Heidelberg, 2008. 550-565.

Techniques Summary

TABLE 3
Different Approaches to Transfer Learning

Transfer Learning Approaches	Brief Description
Instance-transfer	To re-weight some labeled data in the source domain for use in the target domain [6], [28], [29], [30], [31], [24], [32], [33], [34], [35].
Feature-representation-transfer	Find a "good" feature representation that reduces difference between the source and the target domains and the error of classification and regression models [22], [36], [37], [38], [39], [8], [40], [41], [42], [43], [44].
Parameter-transfer	Discover shared parameters or priors between the source domain and target domain models, which can benefit for transfer learning [45], [46], [47], [48], [49].
Relational-knowledge-transfer	Build mapping of relational knowledge between the source domain and the target domains. Both domains are relational domains and i.i.d assumption is relaxed in each domain [50], [51], [52].

TABLE 4
Different Approaches Used in Different Settings

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

Related Problems

- Recognize the limit of the power of transfer learning (transfer bound)
  - Using Kolmogorov complexity[Mahmud07]
  - Graph-based method [Eaton08]
- Negative transfer
  - How to avoid negative transfer automatically?
    - Bayesian approach
    - etc.

Reference

- Pan, Sinno Jialin, and Qiang Yang. "A survey on transfer learning." *Knowledge and Data Engineering, IEEE Transactions on* 22.10 (2010): 1345-1359.
- Mahmud, M. M., and Sylvian Ray. "Transfer learning using Kolmogorov complexity: basic theory and empirical evaluations." *Advances in neural information processing systems*. 2007.
- Eaton, Eric, and Terran Lane. "Modeling transfer relationships between learning tasks for improved inductive transfer." *Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg, 2008. 317-332.