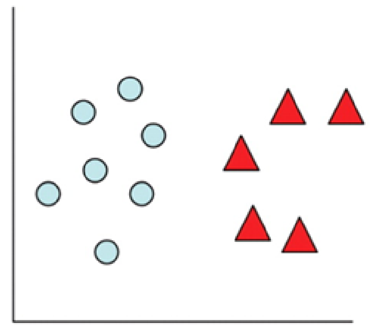
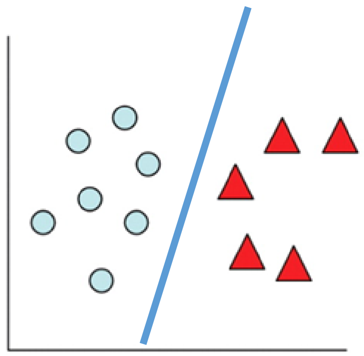
Transfer Learning and
DOMAIN ADAPTATION METHODS

Prof. SUKHENDU DAS
Deptt. of CS&E, IIT Madras
Email: sdas@cse.iitm.ac.in
//www.cse.iitm.ac.in/~vplab

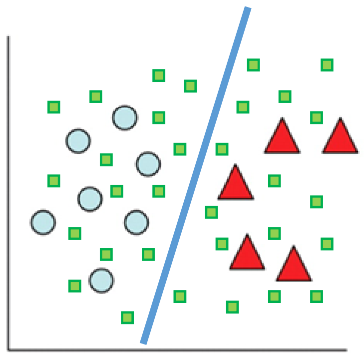
Pattern Classification



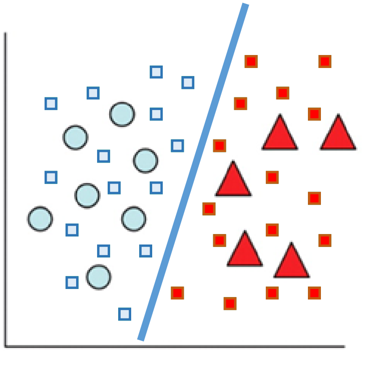
Pattern Classification



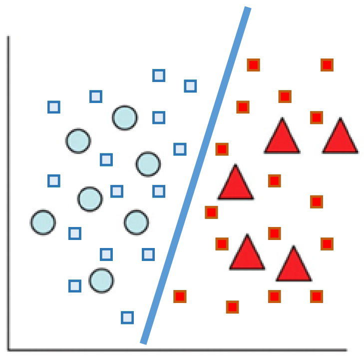
Pattern Classification



Pattern Classification



Pattern Classification

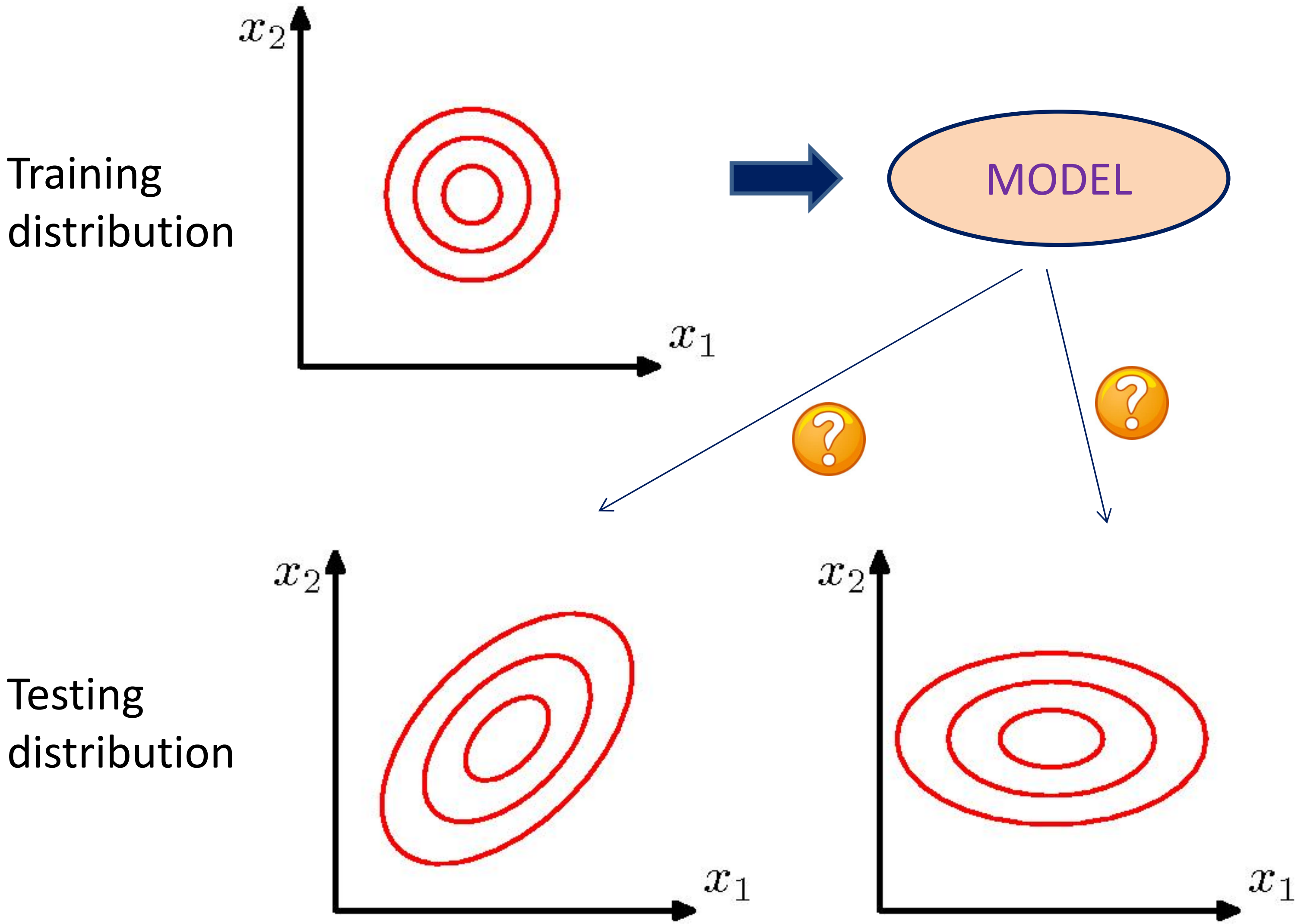


Assumptions - Data are independently and identically distributed

A Major Assumption till now…

Training and future (test) data come from a same task and a same domain.

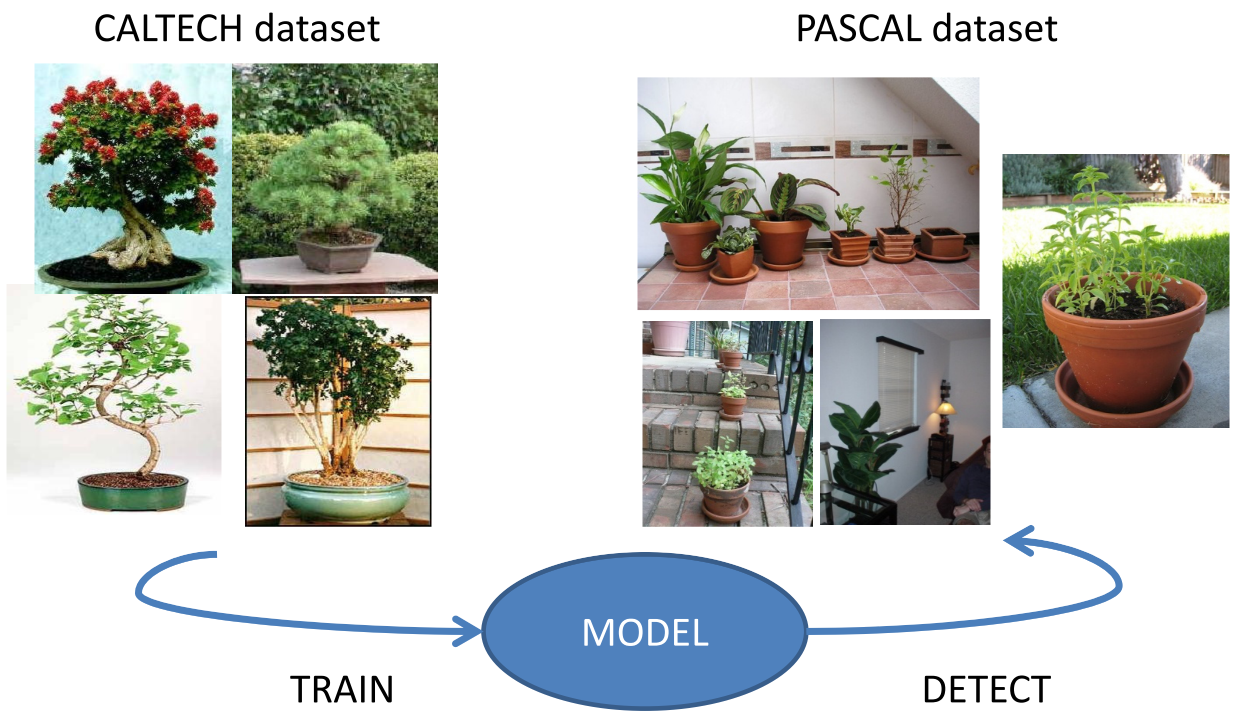
- *Represented in same feature and label spaces.*
- *Follow a same distribution.*


Training distribution
MODEL
x_2
x_1
Testing distribution
x_2
x_1
x_2
x_1

Transfer Learning

In the machine learning community

- The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks or new domains, which share some commonality.



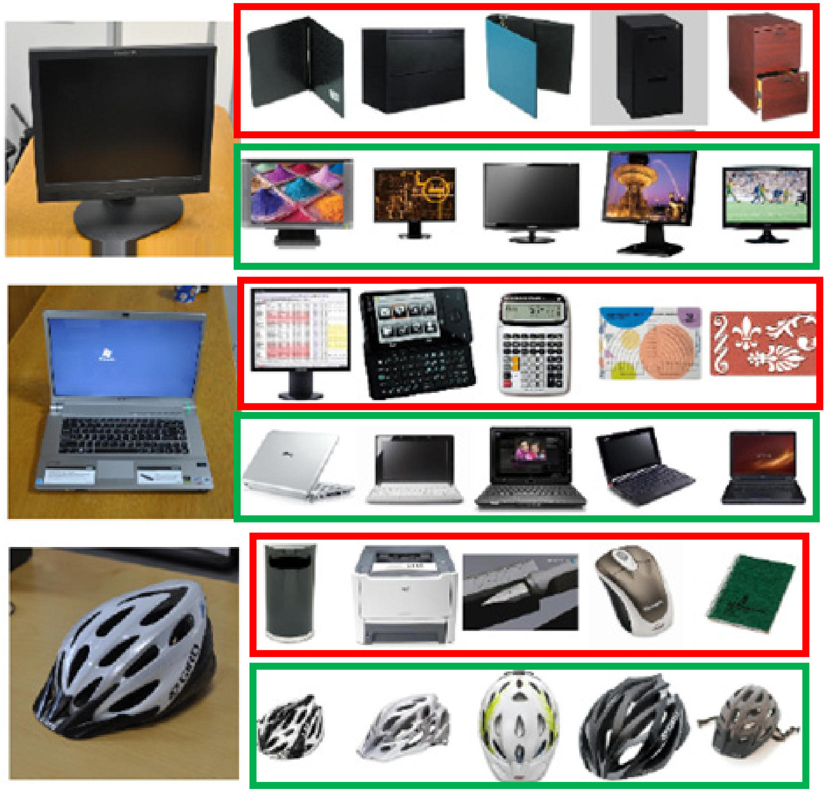
Transfer Learning

- The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks or new domains, which share some commonality.

Labeled data from source domain present.

source domain target domain

Transfer learning uses either small number of labeled data or unlabeled data from target domain.



[Saenko et al. ECCV 10]

Why Transfer Learning?

- In some domains, labeled data are in short supply.
- In some domains, the labeling cost is very expensive.
- In some domains, the learning process is time consuming.

- ***How to extract knowledge*** *learnt from related domains to help learning in a target domain with a few labeled data?*
- ***When to transfer knowledge*** *learnt from the related domain to help the task in the target domain?*

Transfer learning techniques may help!

What is TL:

A major assumption in many machine learning and data mining algorithms is that the training and future data must be in the same feature space and have the same distribution.

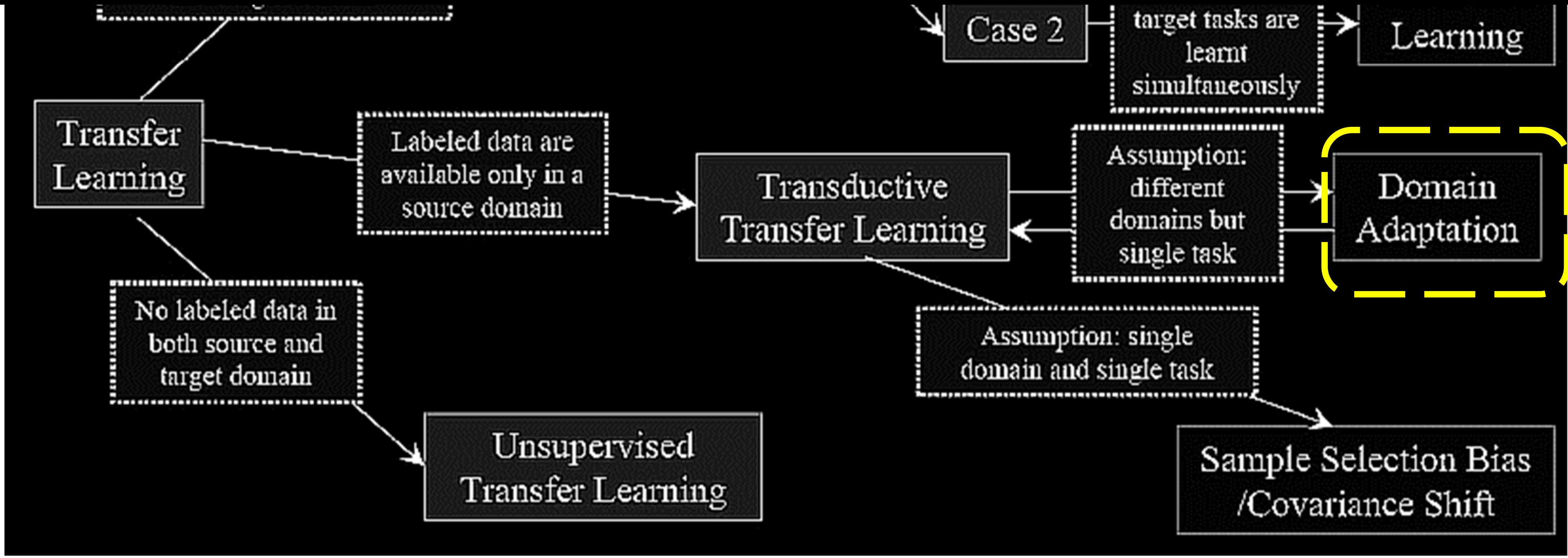
However, in many real-world applications, this assumption may not hold. For example, we sometimes have a classification task in one domain of interest, but we only have sufficient training data in another domain of interest, where the latter data may be in a different feature space or follow a different data distribution.

In such cases, *knowledge transfer,* if done successfully, would greatly improve the performance of learning by avoiding much expensive data-labeling efforts.

Definition 1 (Transfer Learning). *Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$,* transfer learning *aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.*

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



Approaches to Transfer Learning

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	√		

Model-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-transfer	√		
Relational-knowledge-transfer	√		

TL applications:

sensor-network-based localization, text classification, image classification, video classification, social network analysis, and logical inference.

Definition 3 (Transductive Transfer Learning). *Given a source domain* $\mathcal{D}_S$ *and a corresponding learning task* $\mathcal{T}_S$*, a target domain* $\mathcal{D}_T$ *and a corresponding learning task* $\mathcal{T}_T$*,* transductive transfer learning *aims to improve the learning of the target predictive function* $f_T(\cdot)$ *in* $\mathcal{D}_T$ *using the knowledge in* $\mathcal{D}_S$ *and* $\mathcal{T}_S$*, where* $\mathcal{D}_S \neq \mathcal{D}_T$ *and* $\mathcal{T}_S = \mathcal{T}_T$*. In addition, some unlabeled target-domain data must be available at training time.*

Definition 1 (Transfer Learning). *Given a source domain* $\mathcal{D}_S$ *and learning task* $\mathcal{T}_S$*, a target domain* $\mathcal{D}_T$ *and learning task* $\mathcal{T}_T$*,* transfer learning *aims to help improve the learning of the target predictive function* $f_T(\cdot)$ *in* $\mathcal{D}_T$ *using the knowledge in* $\mathcal{D}_S$ *and* $\mathcal{T}_S$*, where* $\mathcal{D}_S \neq \mathcal{D}_T$*, or* $\mathcal{T}_S \neq \mathcal{T}_T$*.*

Sl N0	TTL - Method	References (not exhaustive)
1	Kernel-mean matching (**KMM**) in RKHS	J. Huang, A. Smola, A. Gretton, K.M. Borgwardt, and B. Scho¨lkopf, "Correcting Sample Selection Bias by Unlabeled Data," Proc. 19th Ann. Conf. Neural Information Processing Systems (NIPS), 2007.
2	Kullback-Leibler Importance Estimation Procedure (**KLIEP**)	M. Sugiyama, S. Nakajima, H. Kashima, P.V. Buenau, and M. Kawanabe, "Direct Importance Estimation with Model Selection and its Application to Covariate Shift Adaptation," Proc. 20th Neural Information Processing Systems, (NIPS) Dec. 2008.
3	Importance sampling and reweighting methods for covariate shift or sample selection bias,	] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N.D. Lawrence, Dataset Shift in Machine Learning. MIT Press, 2009.
4	Structural correspondence learning (SCL)	R.K. Ando and T. Zhang, "A High-Performance Semi-Supervised Learning Method for Text Chunking," Proc. 43rd Ann Meeting on Assoc. for Computational Linguistics, pp. 1-9, 2005.
5	Topic-bridged PLSA (probabilistic latent semantic analysis), or TPLSA	G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-Bridged PLSA for Cross-Domain Text Classification," Proc. 31st Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 627-634, July 2008.
6	Maximum Mean Discrepancy Embedding (**MMDE**)	S.J. Pan, J.T. Kwok, and Q. Yang, "Transfer Learning via Dimensionality Reduction," Proc. 23rd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence, pp. 677-682, July 2008.
7	Transfer Component Analysis (**TCA**)	S.J. Pan, I.W. Tsang, J.T. Kwok, and Q. Yang, "Domain Adaptation via Transfer Component Analysis," Proc. 21st Int'l Joint Conf. Artificial Intelligence, 2009.

Notations

Domain:

- Feature space $\mathcal{X}$;
- $P(x)$, where $x \in \mathcal{X}$.

Two domains are different $\Rightarrow$

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

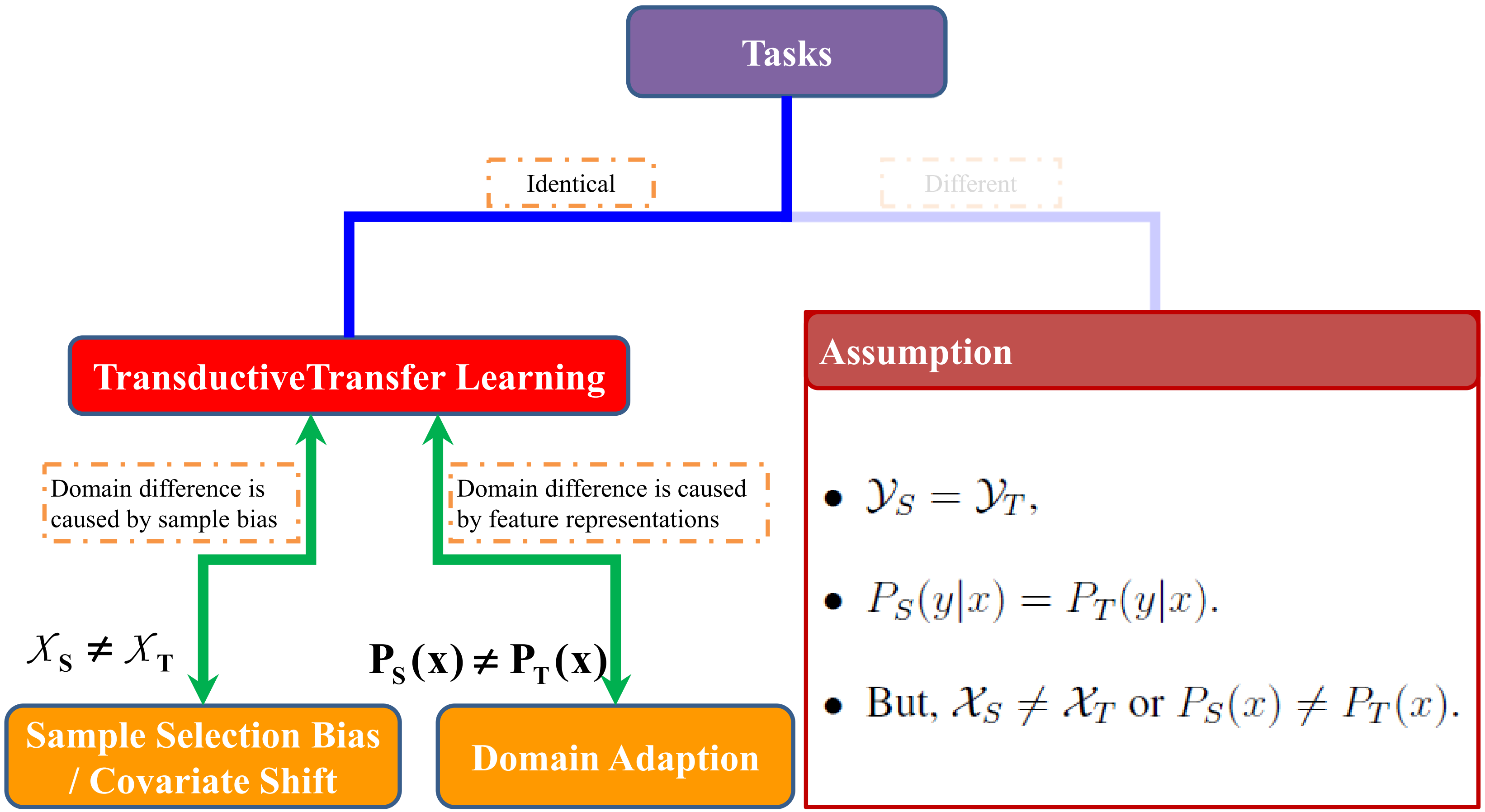
Task:

- Given $\mathcal{X}$ and label space $\mathcal{Y}$;
- To learn $f : x \rightarrow y$, or estimate $P(y|x)$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

Two tasks are different $\Rightarrow$

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

Transductive Transfer Learning



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.

$$\mathbf{P_s(y \mid x) = P_t(y \mid x)}$$

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

$$\mathbf{P_S(x, y) \neq P_T(x, y)}$$

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 / Covariate Shift
[Quionero-Candela, *et al,* Data Shift in Machine Learning, MIT Press 2009]

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.

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

To match means between training and test data in a RKHS

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

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.

SCL

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

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

Output: predictor $f : X \to Y$

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

2. For $\ell = 1$ to m

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

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 $\left\{ \left(\begin{bmatrix} \mathbf{x}_t \\ \theta \mathbf{x}_i \end{bmatrix}, y_t \right)_{t=1}^T \right\}$

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

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

Learn parameters of each prediction problem

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

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

Domain Adaptation
– A type of Transfer Learning

- Domain adaptation of statistical classifiers is the problem that arises when the data distribution in our test domain is different from that in our training domain [Jing Jiang, 2008].

> ◈ ***How to extract knowledge*** *learnt from related domains to help learning in a target domain with a few labeled data?*
>
> ◈ ***When to transfer knowledge*** *learnt from the related domain to help the task in the target domain?*

- ➢ In some domains, labeled data are in short supply.
- ➢ In some domains, the labeling cost is very expensive.
- ➢ In some domains, the learning process is time consuming.

Why Domain Adaptation?

Training samples of two classes

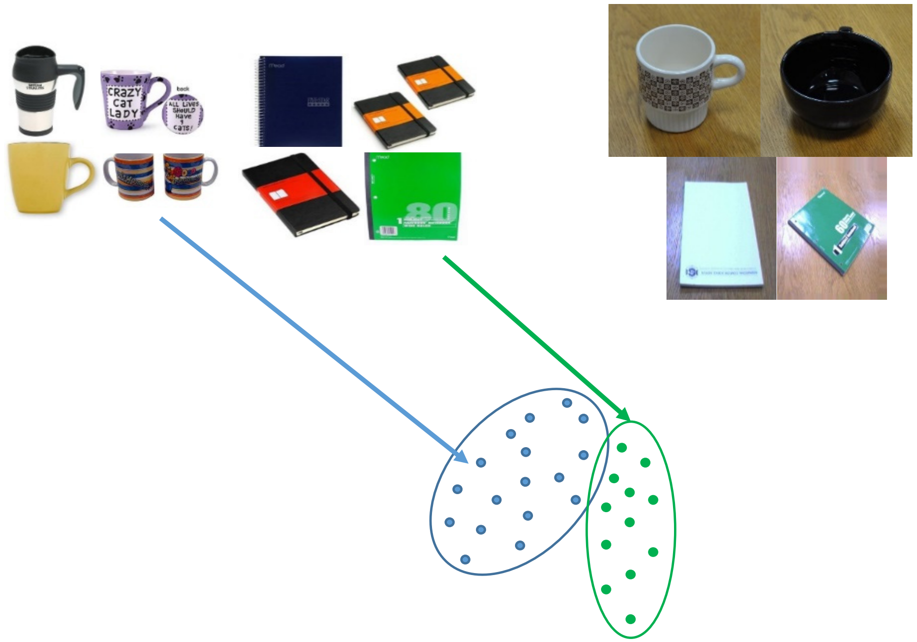
Why Domain Adaptation?

Training samples of two classes

Testing samples of two classes

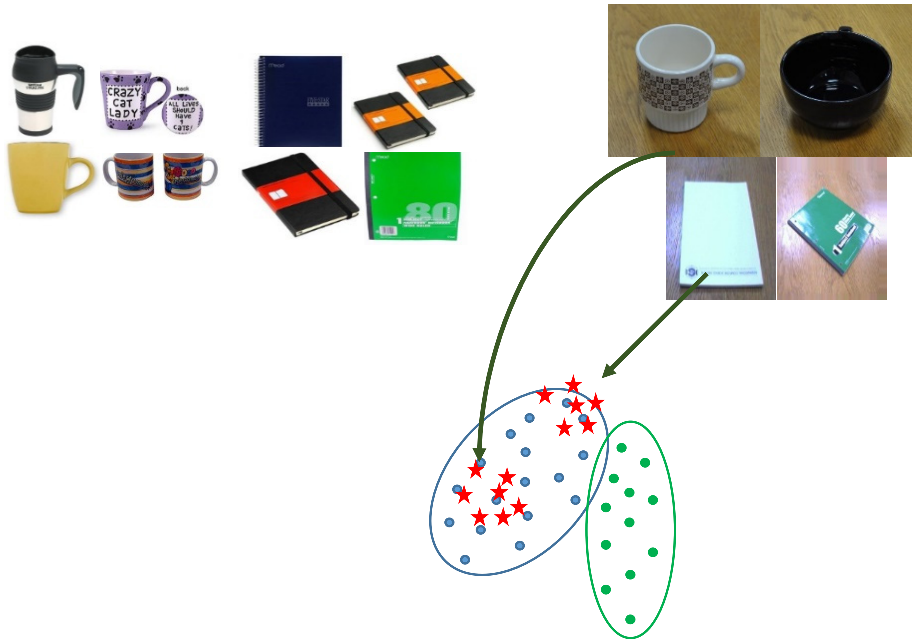
- Training Data is not uniformly sampled.
- Change in the sensor alters the distribution of data.

Why Domain Adaptation?



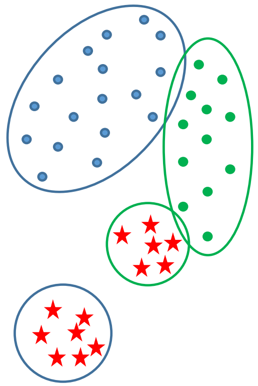
- Training Data is not uniformly sampled.
- Change in the sensor alters the distribution of data.

Why Domain Adaptation?



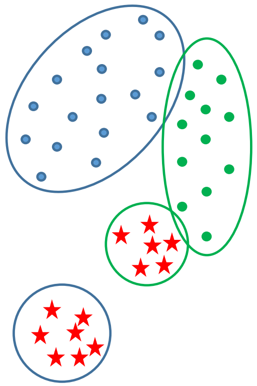
- Training Data is not uniformly sampled.
- Change in the sensor alters the distribution of data.

Why Domain Adaptation?



- Training Data is not uniformly sampled.
- Change in the sensor alters the distribution of data.

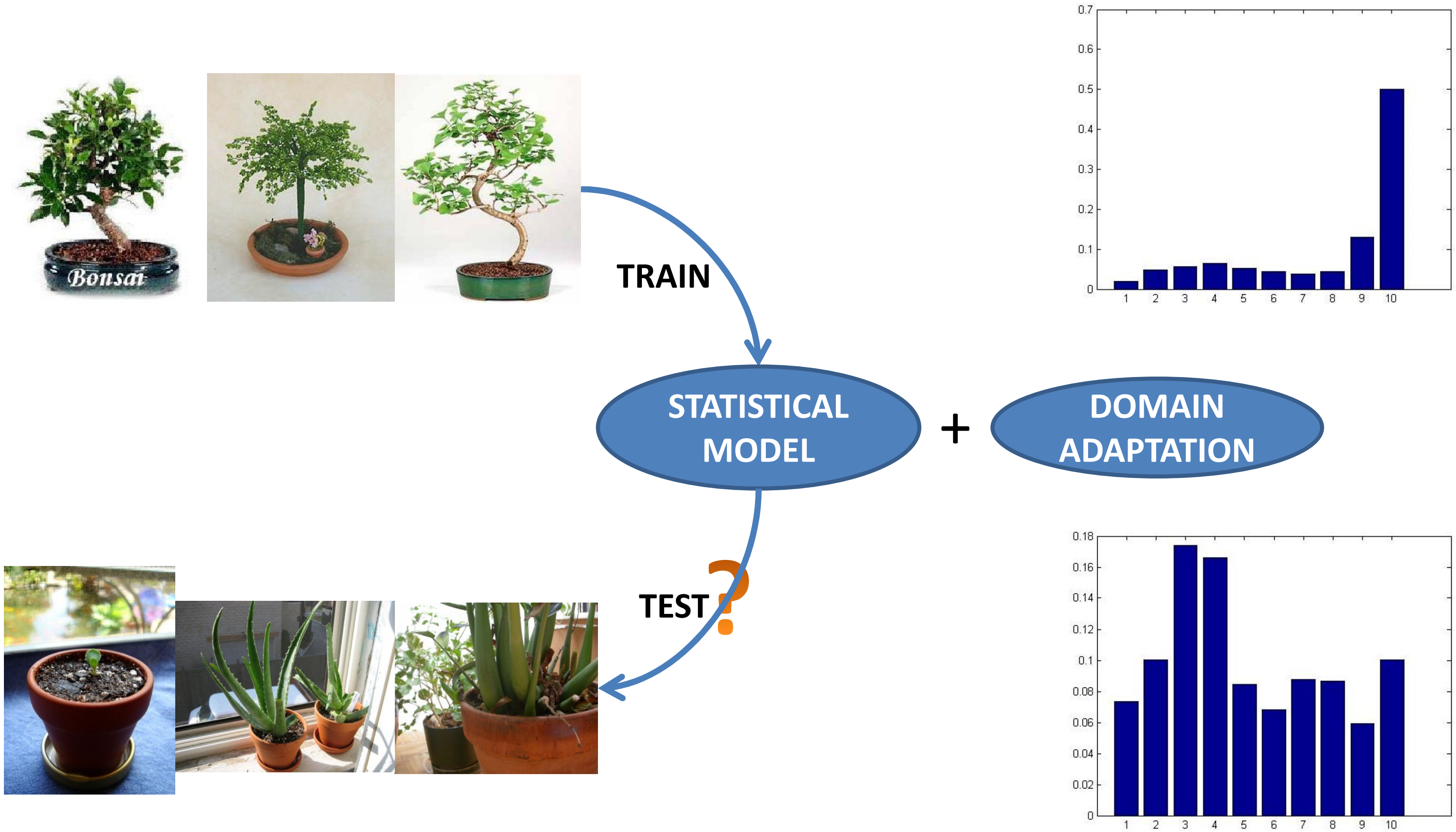
Why Domain Adaptation?



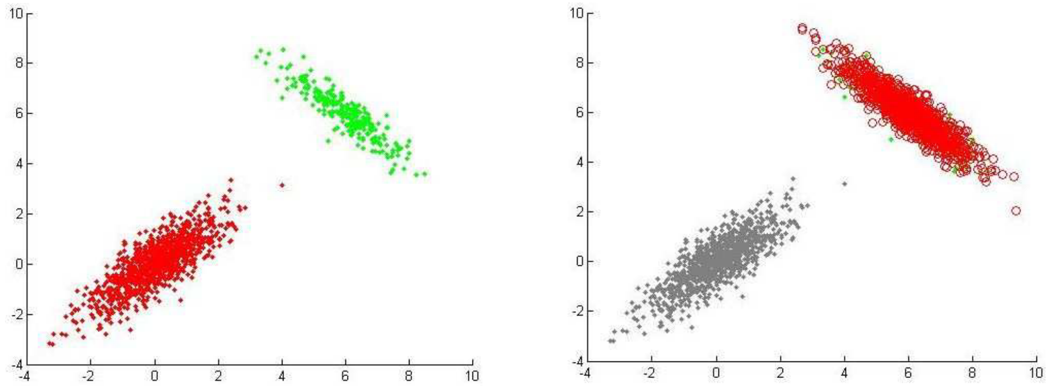
- Training Data is not uniformly sampled.
- Change in the sensor alters the distribution of data.
- Training samples are drawn from **source domain,** and test samples are drawn from **target domain**.

Domain adaptation is the process where one can use the training samples available from source domain to aid a classification task.

An Example



Domain Adaptation (DA)



Source Domain : Gallery Samples
Target Domain : Probe Samples

- Reasons for domain adaptation
  - Difference in resolution
  - Blur
  - Noise
  - Low-contrast
  - Different camera parameters

Common Approaches

- Spatial topology of the instances in source domain is preserved – important parameter for many of the classifiers like KNN, clustering algorithms.
- This ensures a set of constraints for forming the transformation matrix.

- Need to capture the distribution of Target Domain
- Problem: Small number of samples lead to erroneous parameterization of distribution

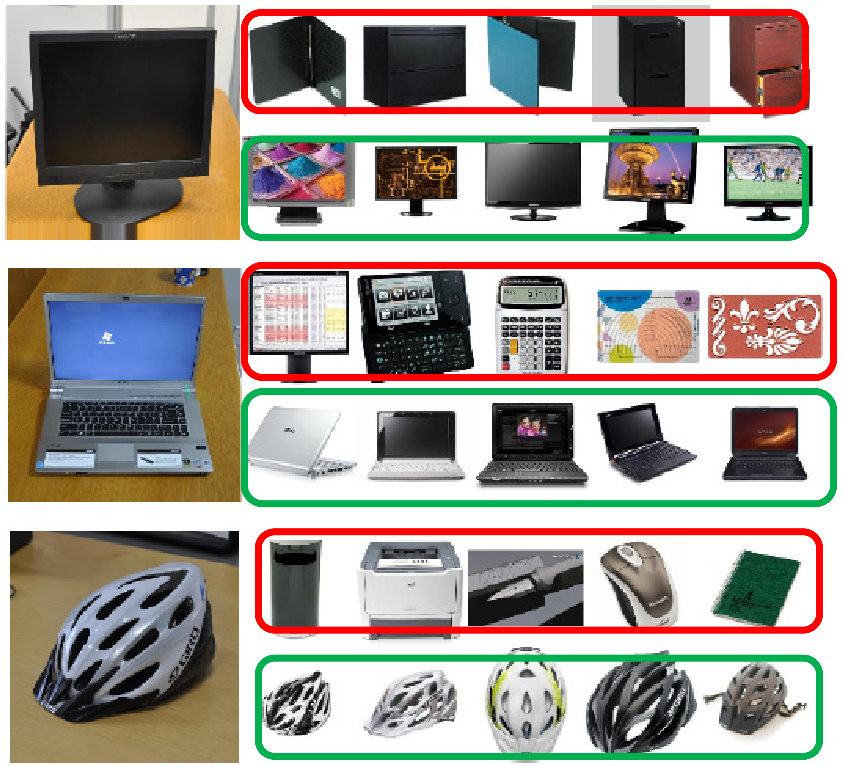
Domain Adaptation

[Saenko et al. ECCV 10]

source domain

target domain

$$\min_W \; r(W)$$
$$\text{s.t.} \quad c_i(X^T W Y) \geq 0, \quad 1 \leq i \leq c.$$



Domain

$X \in Souce\ Domain$

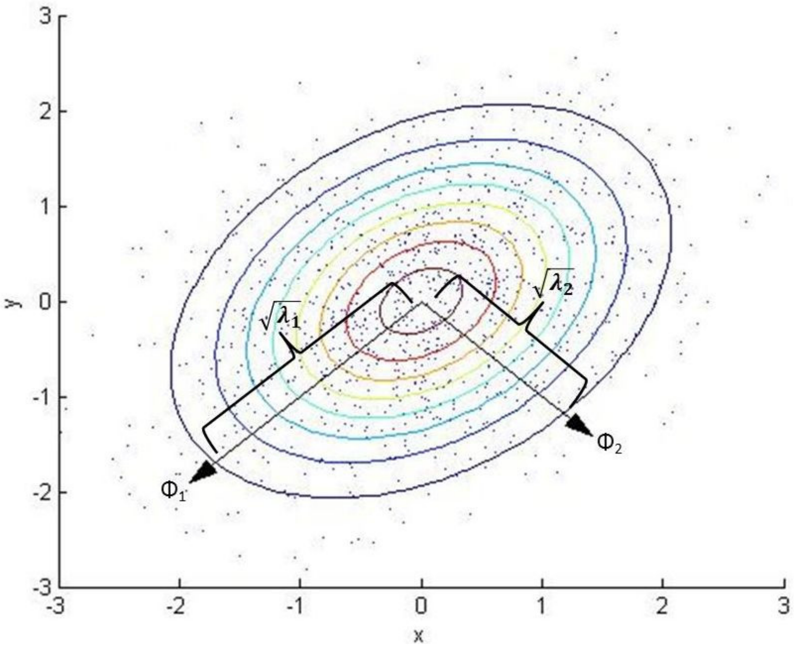
$Y \in Target\ Domain$

$$\mathrm{sim}_W(x, y) = x^T W y.$$

W is the transformation matrix

METHOD 1:
DA BY EIGEN DOMAIN TRANSFORMATION

Nice Property of Gaussian Distribution



$\lambda_i, i = 1,2, \dots, d$ are the set of eigen-values

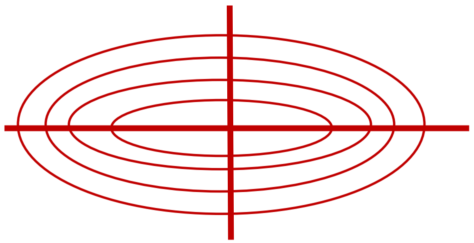
$\Phi_i, i = 1,2, \dots, d$ are the set of eigen-vectors.

PROBLEM: Real world dataset hardly follow a Gaussian Distribution.
SOLUTION: Fit a Gaussian Mixture Model separately in both the domains.

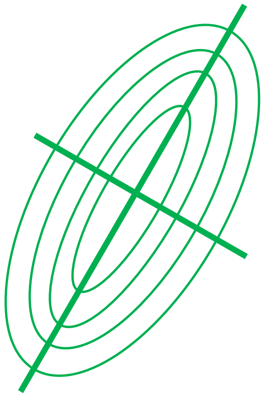
PROBLEM: Small sample size in Target domain.
SOLUTION: Using centroid-based clustering technique to form clusters following Gaussian distribution simultaneously in both the domains.

Eigen Domain Transformation (EDT)

SOURCE DOMAIN

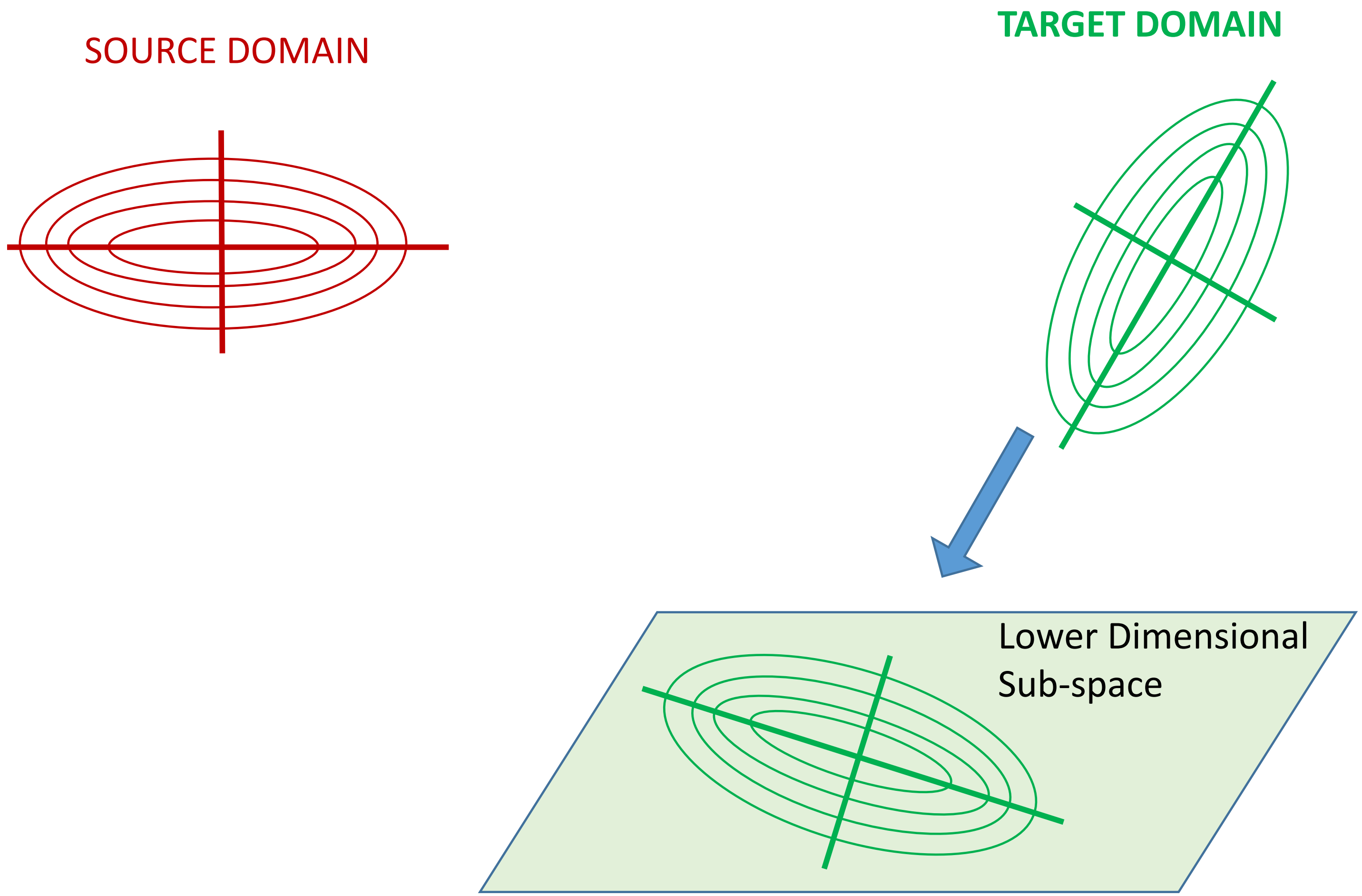


TARGET DOMAIN



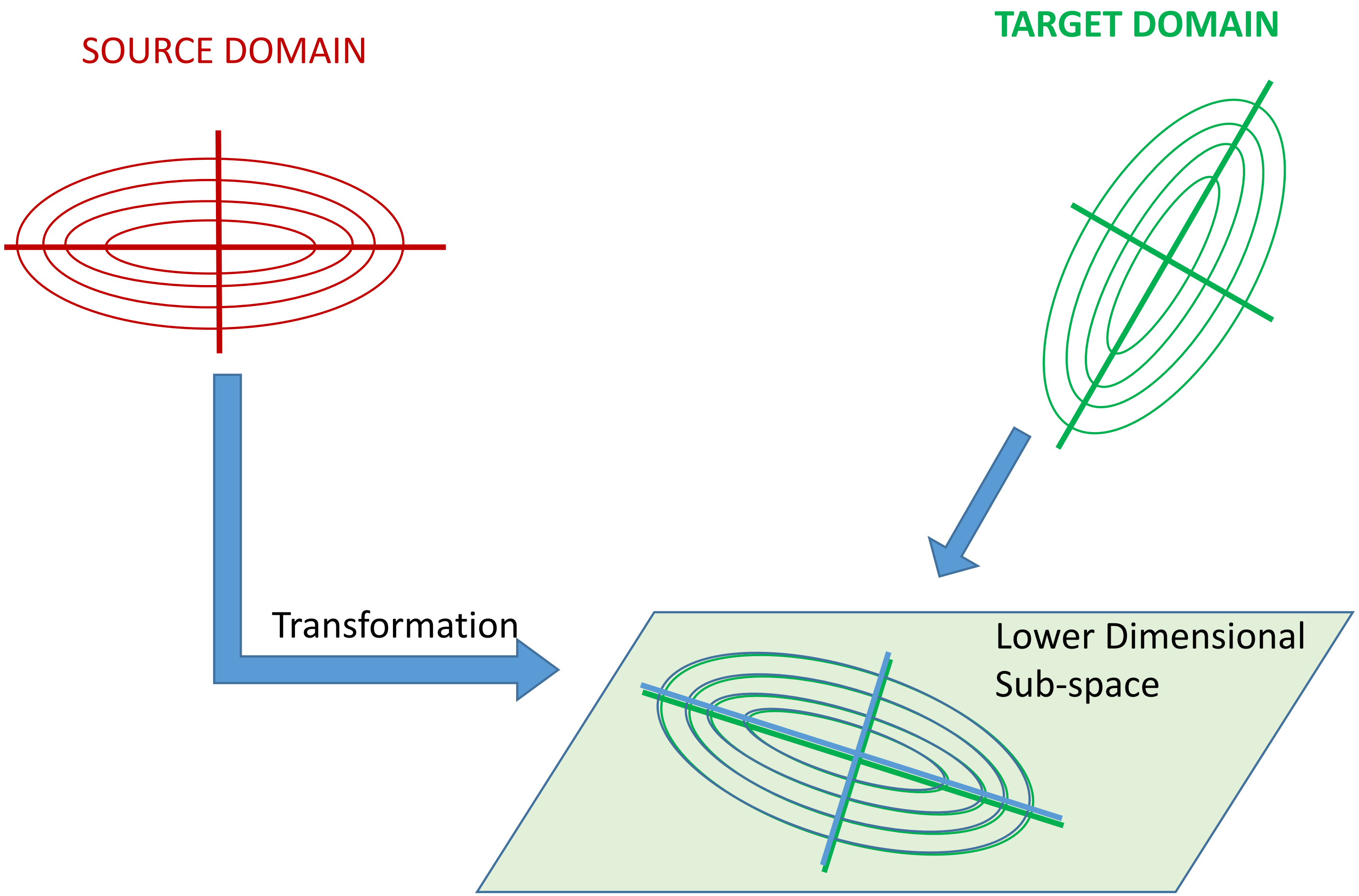
IET-IP (2015)

Eigen Domain Transformation (EDT)



IET-IP (2015)

Eigen Domain Transformation (EDT)



IET-IP (2015)

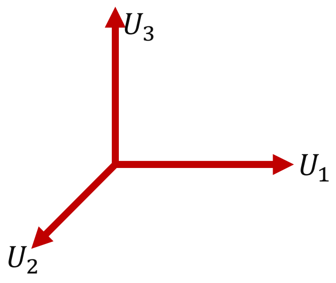
Eigen Domain Transformation (EDT)

- **Finding the optimal number of dimension for estimating sub-space**

IET-IP (2015)

Eigen Domain Transformation (EDT)

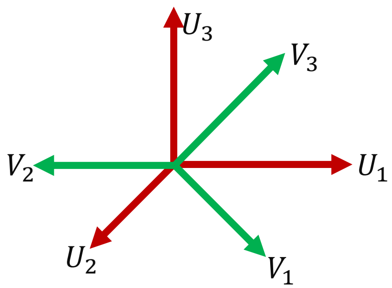
- **Finding the optimal number of dimension for estimating sub-space**



IET-IP (2015)

Eigen Domain Transformation (EDT)

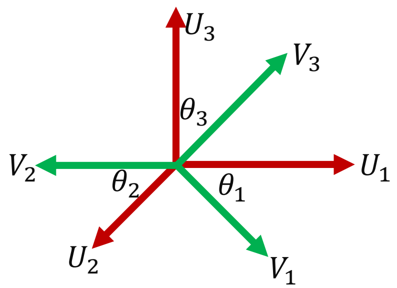
- **Finding the optimal number of dimension for estimating sub-space**



IET-IP (2015)

Eigen Domain Transformation (EDT)

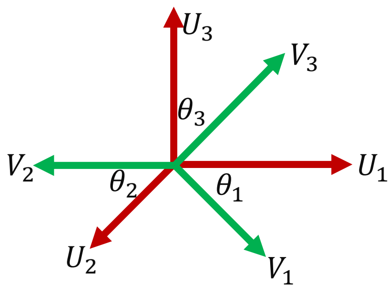
- **Finding the optimal number of dimension for estimating sub-space**



IET-IP (2015)

Eigen Domain Transformation (EDT)

- **Finding the optimal number of dimension for estimating sub-space**



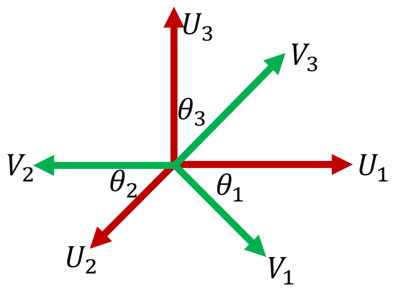
Distance between two sub-spaces

$$\delta_{proj}^2\left(U_p, V_p\right) = p - trace(V_p^T U_p U_p^T V_p)$$

IET-IP (2015)

Eigen Domain Transformation (EDT)

- **Finding the optimal number of dimension for estimating sub-space**



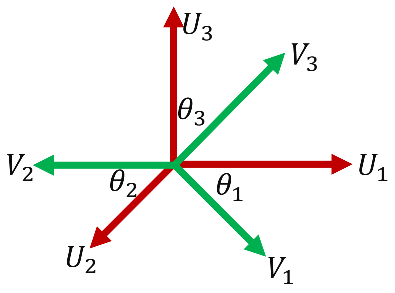
Distance between two sub-spaces

$$\delta_{proj}^2(U_p, V_p) = p - \boldsymbol{trace}(\boldsymbol{V_p^T U_p U_p^T V_p})$$

IET-IP (2015)

Eigen Domain Transformation (EDT)

- **Finding the optimal number of dimension for estimating sub-space**



Distance between two sub-spaces:

$$\delta^2_{proj}(U_p, V_p) = p - trace(V_p^T U_p U_p^T V_p)$$

- **Transformation of source domain data**

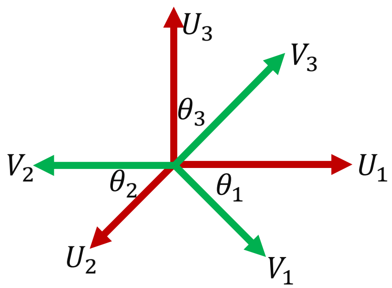
$$\tilde{X} = X U_{p*} \Lambda_{p*}^{-1/2} \Gamma_{p*}^{1/2} V_{p*}^T$$

Extension to RKHS has been proposed Non-linear Transformation

IET-IP '15

Eigen Domain Transformation (EDT)

- **Finding the optimal number of dimension for estimating sub-space**



Distance between two sub-space

$$\delta^2_{proj}\left(U_p, V_p\right) = p - trace(V_p^T U_p U_p^T V_p)$$

- **Transformation of source domain data**

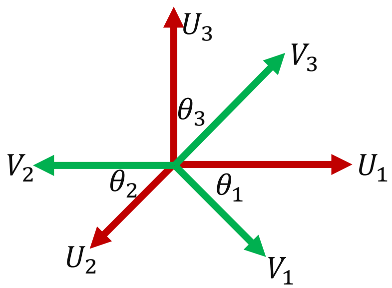
$$\tilde{X} = XU_{p*}\Lambda_{p*}^{-1/2}\Gamma_{p*}^{1/2}V_{p*}^T$$

Eigen-vector of source domain

Extension to RKHS has been proposed Non-linear Transformation

Eigen Domain Transformation (EDT)

- **Finding the optimal number of dimension for estimating sub-space**



Distance between two sub-space

$$\delta_{proj}^2\left(U_p, V_p\right) = p - trace(V_p^T U_p U_p^T V_p)$$

- **Transformation of source domain data**

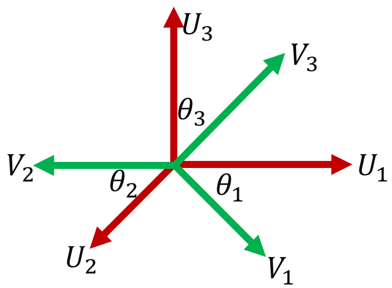
$$\tilde{X} = XU_{p*}\Lambda_{p*}^{-1/2}\Gamma_{p*}^{1/2}V_{p*}^T$$

Eigen-value of source domain

Extension to RKHS has been proposed Non-linear Transformation

Eigen Domain Transformation (EDT)

- **Finding the optimal number of dimension for estimating sub-space**



Distance between two sub-spaces

$$\delta^2_{proj}\left(U_p, V_p\right) = p - trace(V_p^T U_p U_p^T V_p)$$

- **Transformation of source domain data**

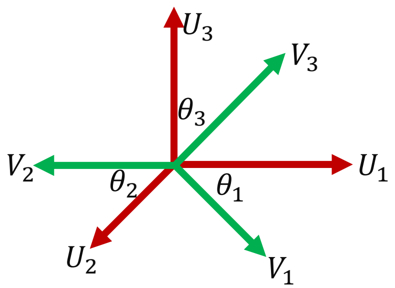
$$\tilde{X} = XU_{p*}\Lambda_{p*}^{-1/2}\Gamma_{p*}^{1/2}V_{p*}^T$$

Eigen-
value of
target
domain

Extension to RKHS has been proposed
Non-linear Transformation

Eigen Domain Transformation (EDT)

- **Finding the optimal number of dimension for estimating sub-space**



Distance between two sub-space

$$\delta^2_{proj}(U_p, V_p) = p - trace(V_p^T U_p U_p^T V_p)$$

- **Transformation of source domain data**

$$\tilde{X} = XU_{p*}\Lambda_{p*}^{-1/2}\Gamma_{p*}^{1/2}V_{p*}^T$$

Eigen-vector of target domain

Extension to RKHS has been proposed Non-linear Transformation

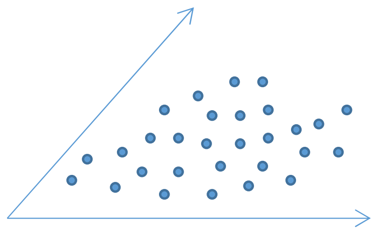
METHOD 2:
DA USING DOMAIN INVARIANT FEATURES

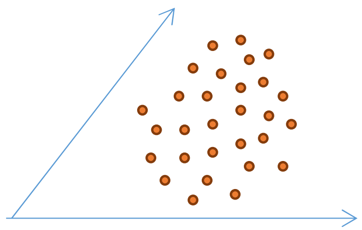
Discrepancy between Distributions

- **Maximum Mean Discrepancy (MMD):** The distance between a pair of distributions can be well estimated by the distance of the means of the two samples in Reproducing Kernel Hilbert Space (RKHS) [Gretton 2009, Pan et al 2009].

Discrepancy between Distributions

Source: X

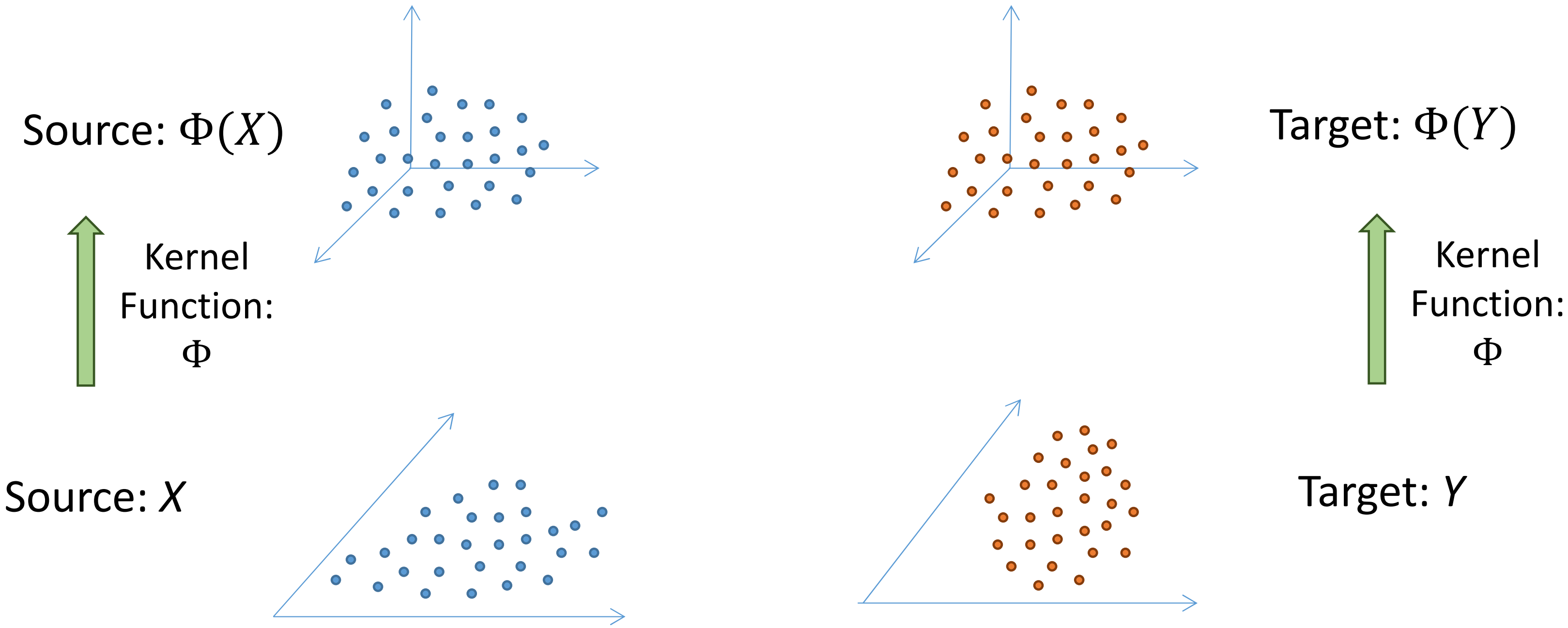




Target: Y

- **Maximum Mean Discrepancy (MMD):** The distance between a pair of distributions can be well estimated by the distance of the means of the two samples in Reproducing Kernel Hilbert Space (RKHS) [Gretton 2009, Pan et al 2009].

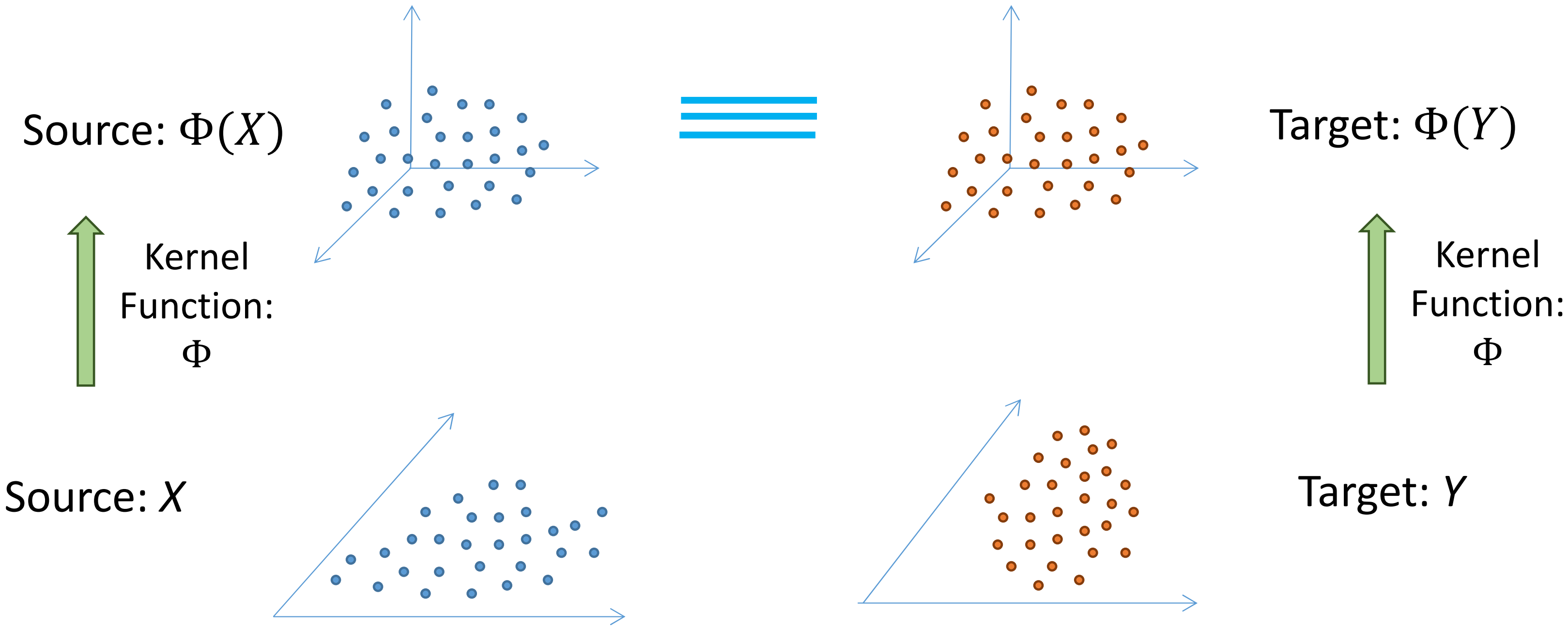
Discrepancy between Distributions



- **Maximum Mean Discrepancy (MMD):** The distance between a pair of distributions can be well estimated by the distance of the means of the two samples in Reproducing Kernel Hilbert Space (RKHS) [Gretton 2009, Pan et al 2009].

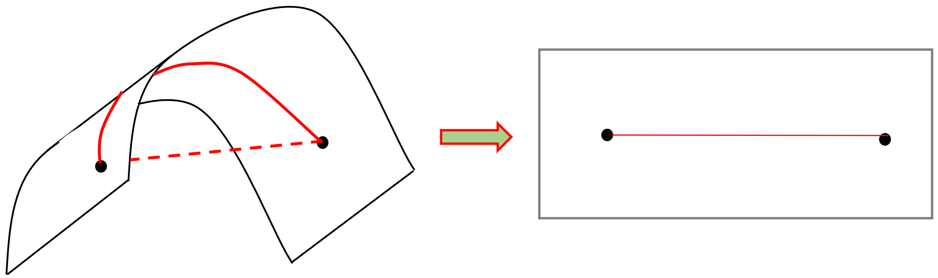
Discrepancy between Distributions

Mean of $\Phi(X)$ = Mean of $\Phi(Y)$



- **Maximum Mean Discrepancy (MMD):** The distance between a pair of distributions can be well estimated by the distance of the means of the two samples in Reproducing Kernel Hilbert Space (RKHS) [Gretton 2009, Pan et al 2009].

Introducing Manifold



- **Manifold:** A space which locally looks Euclidean

- **AIM: Find a sub-space *W,* where the underlying distributions and manifolds of two domains are same.**

Cost functions

A. Difference in means between two domains:

- Minimize the disparity in distributions of two domains using the concept of MMD

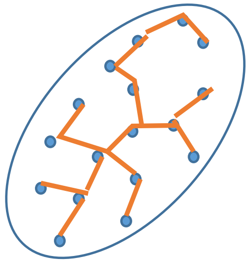
Cost functions

A. Difference in means between two domains:

- Minimize the disparity in distributions of two domains using the concept of MMD

B. Preserving local spatial arrangement of data:

- For an instance in source domain, the set of the instances that forms its neighborhood remains preserved after transformation



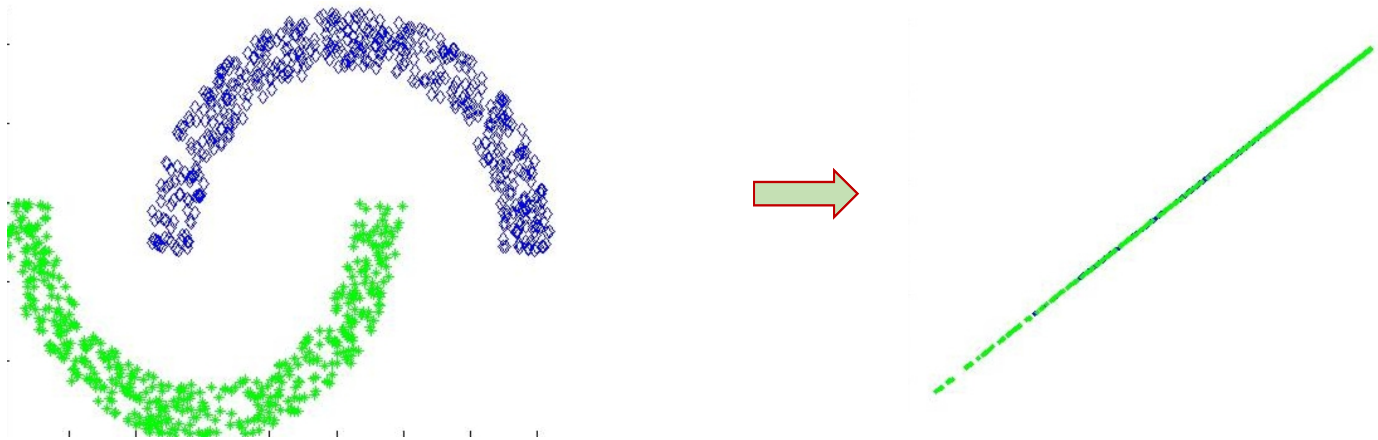
Cost functions

A. Difference in means between two domains:

- Minimize the disparity in distributions of two domains using the concept of MMD

B. Preserving local spatial arrangement of data:

- For an instance in source domain, the set of the instances that forms its neighborhood remains preserved after transformation



ICIP' 14

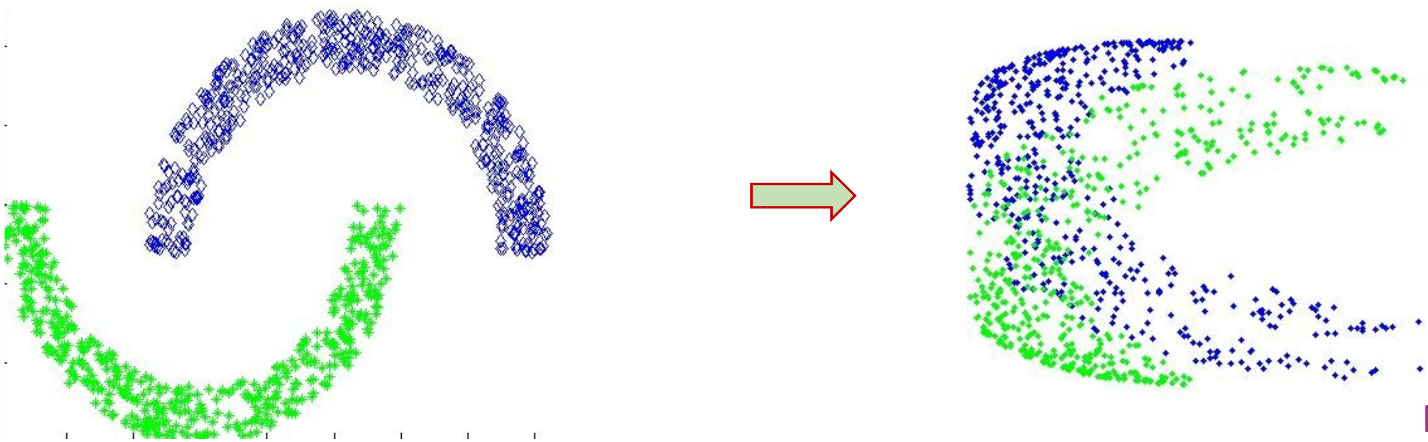
Cost functions

A. Difference in means between two domains:

- Minimize the disparity in distributions of two domains using the concept of MMD

B. Preserving local spatial arrangement of data:

- For an instance in source domain, the set of the instances that forms its neighborhood remains preserved after transformation



ICIP’ 14

Defining Landmark points

- **Manifold distance**: distance between two points lying on a manifold can be approximated by the length of the path between the two points using an adjacency graph.

Jun Li and Pengwei Hao, “Finding representative landmarks of data on manifolds,” Pattern Recognition, 2009

Defining Landmark points

- **Manifold distance**: distance between two points lying on a manifold can be approximated by the length of the path between the two points using an adjacency graph.

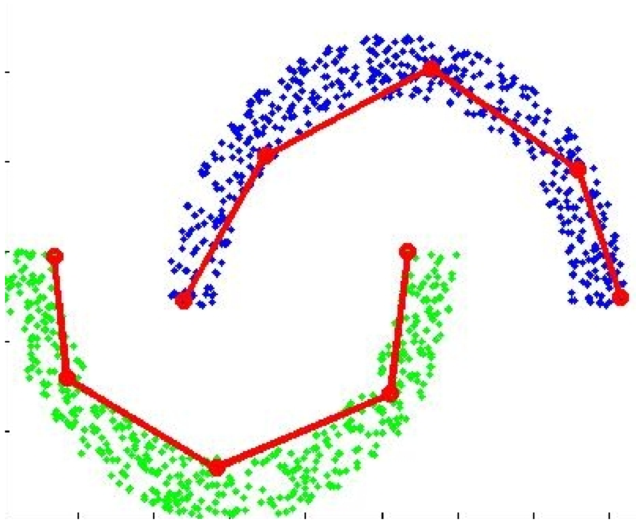
- If x and y are two **landmark points**, then a third landmark point, lying in between them can be defined as:

$$z = \underset{w}{\operatorname{argmax}} \left(\frac{Manifold\ dist.from}{adjacent\ Landmark\ points} - \frac{Euclidean\ dist.from}{adjacent\ Landmark\ points} \right)$$

Jun Li and Pengwei Hao, "Finding representative landmarks of data on manifolds," Pattern Recognition, 2009

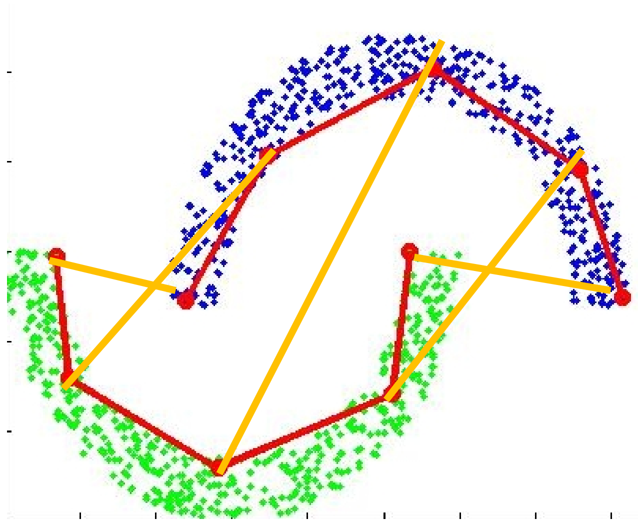
Cost functions

C. Estimating disparity in the shape of the two domains



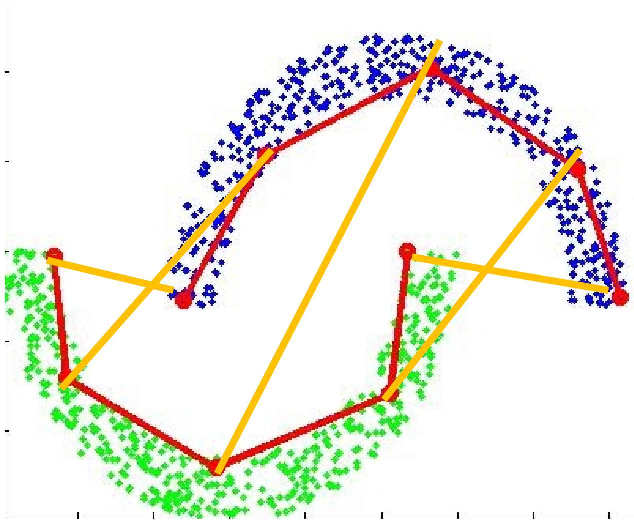
Cost functions

C. Estimating disparity in the shape of the two domains



Cost functions

C. Estimating disparity in the shape of the two domains



D. Inter-class distance in source domain:

Maximize the inter-class distance in source domain in the sub-space to be estimated.

Optimization framework

- **Combined cost function:**

$$f(W) = trace(W^T P W)$$

where $P = [X^T \quad Y^T] \begin{bmatrix} I_1 + I_4 + 2(B_X - A_X) - M & -{I_2}^T - {I_6}^T I_5 \\ -{I_2}^T - {I_6}^T I_5 & I_3 + 2(B_Y - A_y) + I_7 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}$

I_1, I_2 and I_3 - Indicator matrices to calculate difference of means

M – used to estimate inter-class distance in source domain

A_X and B_X - Indicator matrices to represent distance between edges in MST build on source domain

A_Y and B_Y - Indicator matrices to represent distance between edges in MST build on target domain

I_4, I_5, I_6 and I_7 - Indicator matrices to represent sum of distances between corresponding landmark points in two domains.

Optimization framework

- **Combined cost function in RKHS:**

$$f(Z) = trace(Z^T P_K Z)$$

where $P_K = K \begin{bmatrix} I_1 + I_4 + 2(B_X - A_X) - M & -{I_2}^T - {I_6}^T I_5 \\ -{I_2}^T - {I_6}^T I_5 & I_3 + 2\left(B_Y - A_y\right) + I_7 \end{bmatrix} K$

Where, $W = [\Phi(X)^T \quad \Phi(Y)^T]Z$ and $K = \begin{bmatrix} \Phi(X) \\ \Phi(Y) \end{bmatrix} [\Phi(X)^T \quad \Phi(Y)^T]$

Optimization framework

- **Combined cost function in RKHS:**

$$f(Z) = trace(Z^T P_K Z)$$

$$\text{where } P_K = K\begin{bmatrix} I_1 + I_4 + 2(B_X - A_X) - M & -{I_2}^T - {I_6}^T I_5 \\ -{I_2}^T - {I_6}^T I_5 & I_3 + 2\left(B_Y - A_y\right) + I_7 \end{bmatrix}K$$

$$\text{Where, } W = [\Phi(X)^T \quad \Phi(Y)^T]Z \quad \text{ and } K = \begin{bmatrix} \Phi(X) \\ \Phi(Y) \end{bmatrix}[\Phi(X)^T \quad \Phi(Y)^T]$$

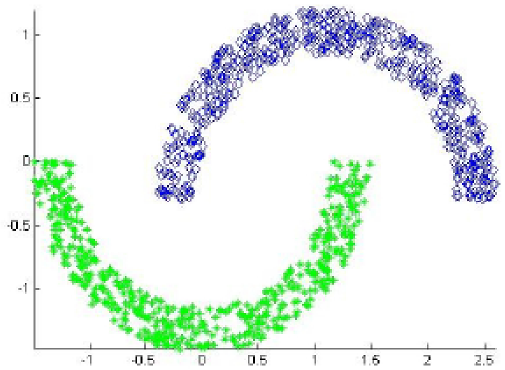
- **Optimization function:**

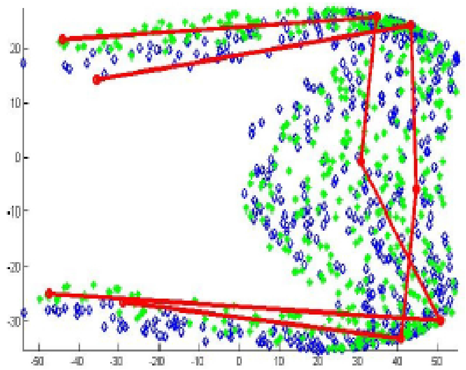
$$\boxed{\begin{array}{ll} minimize & tr(Z^T P_K Z) \\ subject\ \ to & Z^T K Z = I \end{array}}$$

- **Solution:** Z is formed by the Eigenvectors of $P_K^{-1}K$
- If Z_1 and Z_2 are the matrices containing first n_X rows and last n_Y rows of Z, then:
- Transformed Source Domain data: $\boldsymbol{\widetilde{X} = K_{XX}Z_1 + K_{XY}Z_2}$
- Transformed Target Domain data: $\boldsymbol{\widetilde{Y} = K_{XY}^T Z_1 + K_{YY}Z_2}$

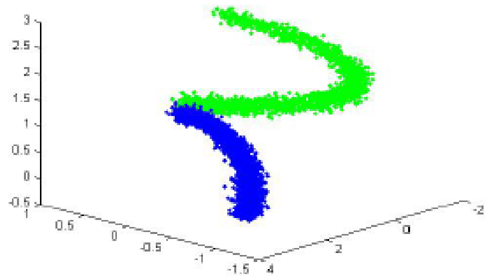
Results: Synthetic (toy) data

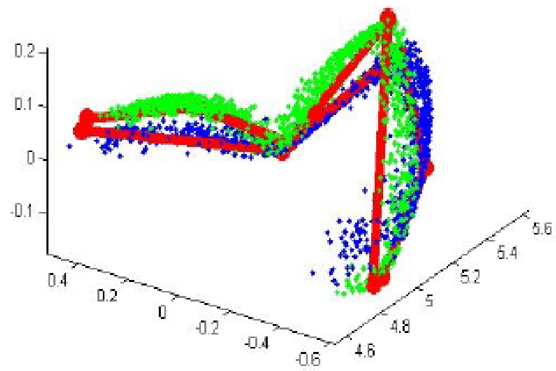
2D example





3D example





Before transformation After transformation

DATASET: Office+Caltech

- Proposed by Saenko et al. (ECCV 2010)
- Extended by Gong et al. (CVPR 2012)

31 classes of objects, 3 domains

10 classes of objects, 4 domains

DATASET: Office+Caltech

- Proposed by Saenko et al. (ECCV 2010)
- Extended by Gong et al. (CVPR 2012)

31 classes of objects, 3 domains

10 classes of objects, 4 domains

Number of training samples per class

Source Domain				Target Domain			
Amazon	**Caltech**	**DSLR**	**Webcam**	**Amazon**	**Caltech**	**DSLR**	**Webcam**
20	20	8	8	3	3	3	3

Experimental results

- Object categorization on Office+Caltech dataset
  - KNN (K=1) classifier has been used

NA – No adaptation

Method	C->A	D->A	W->A	A->C	D->C	W->C	A->D	C->D	W->D	A->W	C->W	D->W	Avg.
NA	21.5	26.9	20.8	22.8	24.8	16.4	22.4	21.7	40.5	23.3	20.0	53.0	26.2
TCA	21.9	16.8	13.4	16.2	17.7	11.1	16.7	22.8	32.3	23.6	22.0	44.7	21.6
GFS	36.9	32.0	27.5	35.3	29.4	21.7	30.7	32.6	54.3	31.0	30.6	66.0	35.7
GFK	36.9	32.5	31.1	35.6	29.8	27.2	35.2	35.2	70.6	34.4	33.7	74.9	39.8
SA	39.0	38.0	37.4	35.3	32.4	32.3	37.6	38.6	80.3	38.6	36.8	83.6	44.2
DA-ClET	35.6	36.2	37.5	34.6	32.7	30.3	33.2	38.7	72.8	35.7	33.6	72.8	41.1
DA-GMCV	34.2	37.3	37.9	36.8	32.2	30.6	34.5	38.9	76.9	36.4	37.1	75.9	42.4
DA-PSA	40.2	39.2	39.4	36.3	33.3	32.7	36.8	40.8	81.2	38.4	36.6	81.4	44.7
DA-EDT	40.7	44.4	48.8	36.6	37.0	36.2	43.3	44.1	85.8	40.0	38.5	85.3	48.2
DA-DIF	56.4	39.8	42.9	48.4	43.8	36.7	39.2	46.6	85.6	39.3	38.4	83.8	50.0

- TCA - S J Pan, I.W. Tsang, J.T. Kwok, and Qiang Yang, “Domain adaptation via transfer component analysis,” IEEE Trans on Neural Networks, 2011.
- GFS - Raghuraman Gopalan, Ruonan Li, and R Chellappa, “Domain adaptation for object recognition: An unsupervised approach,” in ICCV, 2011.
- GFK - Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman, “Geodesic flow kernel for unsupervised domain adaptation,” CVPR, 2012.
- SA - B Fernando, A Habrard, M Sebban, and T Tuytelaars, “Unsupervised visual domain adaptation using subspace alignment,” ICCV, 2013.

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.

Publications

1. "Unsupervised Domain Adaptation using Eigen-Vectors for Object Categorization", Suranjana Samanta and S. Das; **IET Image Processing**, Special issue on Machine Learning for Image Processing, Volume 9, Issue 11, November 2015, pp. 925-930; (**Impact Factor: 1.4**), DOI:10.1049/iet-ipr.2014.0754.
2. "Minimising Disparity in Distribution for Unsupervised Domain Adaptation by Preserving the Local Spatial Arrangement of Data"; Suranjana Samanta and Sukhendu Das**; IET Computer Vision (Impact Factor 1.09),** Volume 10, Issue 5, August 2016, pp. 443-449. DOI:10.1049/iet-cvi.2015.0322.
3. Mutual variation of Information on Transfer-CNN for Face Recognition with degraded probe samples. Samik Banerjee, Sukhendu Das. **Neurocomputing, Elsevier, (Impact Factor: 3.317**), Volume 310, October 2018, pp. 299-315, (May, 2018), DOI: 10.1016/j.neucom.2018.05.038.
4. Soft-Margin Learning for Multiple Feature-Kernel Combinations With Domain Adaptation, for Recognition in Surveillance Face Dataset. Samik Banerjee, Sukhendu Das. Proceedings of 29th CVPR (**CVPRW**) Workshop on Biometrics, IEEE, (Google h5-index: 45) pp. 169-174, Las Vegas, USA (June, 2016); #Ctn – 6
5. Face Recognition in Surveillance Conditions with Bag-of-words, using Unsupervised Domain Adaptation. Samik Banerjee, Sukhendu Das. Proceedings of 9th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), ACM, pp. 50, IISc. Bangalore, India, (December, 2014); #Ctn – 10.
6. "Unsupervised Domain Adaptation Using Manifold Alignment for Object and Event Categorization", S. Samanta and S. Das, in International Conference on Image Processing **(ICIP**), France, 2014.
7. "Modeling Sequential Domain Shift through Estimation of Optimal Sub-spaces for Categorization", S. Samanta, T. Selvan and S. Das, in British Machine Vision Conference **(BMVC**), UK, 2014.
8. "Domain Adaptation Based on Eigen-Analysis and Clustering, for Object Categorization", S. Samanta and S. Das, in International Conference on Computer Analysis of Images and Patterns (CAIP), UK, 2013.

References

- **S. J. Pan, I. Tsang, J. Kwok, and Q. Yang, “Domain adaptation via transfer component analysis”, IEEE Transactions on Neural Networks, 2011.**
- **M. Baktashmotlagh, M. Harandi, B. Lovell, and M. Salzmann, ”Unsupervised domain adaptation by domain invariant projection”, in ICCV, 2013.**
- **A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Scholkopf, “Covariate shift by kernel mean matching”, Dataset shift in machine learning, 2009.**
- **C. Wang and S. Mahadevan, “Heterogeneous domain adaptation using manifold alignment”, IJCAI, 2011**
- **J. Yang, R. Yan, and A. G. Hauptmann, “Cross-domain video concept detection using adaptive SVMs”, ICM, 2007.**
- **R. Gopalan, R. Li, and R. Chellappa, “Domain adaptation for object recognition: An unsupervised approach”, ICCV, 2011.**
- **B. Gong, Y. Shi, F. Sha, and K. Grauman, “Geodesic flow kernel for unsupervised domain adaptation”, CVPR, 2012.**
- **B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, “Unsupervised visual domain adaptation using subspace alignment”, ICCV, 2013.**
- **L. Duan, D. Xu, I. W. Tsang, and J. Luo, “Visual event recognition in videos by learning from web data”, IEEE TPAMI, 2012.**
- **J. Li and P. Hao, “Finding representative landmarks of data on manifolds”, PR, 2009.**
- **D. Zhai, B. Li, H. Chang, S. Shan, X. Chen, and W. Gao, “Manifold alignment via corresponding projections”, BMVC, 2010.**

References

- A. Asuncion, D. N., 2007. UCI machine learning repository; http://www.ics.uci.edu/∼mlearn/MLRepository.html
- Saenko, K., Kulis, B., Fritz, M., Darrell, T., 2010. Adapting visual category models to new domains. In: European conference on Computer Vision. pp. 213–226.
- Chattopadhyay, R., Krishnan, N. C., Panchanathan, S., 2011. Topology preserving domain adaptation for addressing subject based variability in semg signal. In: AAAI Spring Symposium: Computational Physiology.
- Sugiyama, M., Nakajima, S., Kashima, H., von Bu¨nau, P., Kawanabe, M., 2007. Direct importance estimation with model selection and its application to covariate shift adaptation. In: Neural Information Processing Systems. pp. 1962–1965.
- Pan, S. J., Yang, Q., 2010. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22, 1345–1359.
- Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., Sch¨olkopf, B., 2006. Correcting sample selection bias by unlabeled data. In: Neural Information Processing Systems. pp. 601–608.
- **Sinno Jialin Pan;** Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, PPT - **A Survey on Transfer Learning**
- Chang Wang, A Geometric Framework For Transfer Learning Using Manifold Alignment, Ph.D. thesis, University of Massachusetts at Amherst, 2010.