Domain Adaptation for Visual Applications
Part 1: Basic Concepts and Traditional Methods

Mathieu Salzmann

EPFL-CVLab & ClearSpace

Outline

- Basic concepts

- Traditional domain adaptation
  - Metric learning
  - Subspace representations
  - Matching distributions

Basic Concepts

Standard Visual Recognition

Training data

Test data

Train a classifier on the training data and directly apply it to the test data

Domain Shift

Training data

Source domain

Test data

Target domain

A classifier trained on one domain may perform poorly on another domain

Semi-supervised vs Unsupervised

- Semi-supervised: Some labeled target data, but not enough to train from scratch

Source data

Fully-labeled

Target data

A few labels

Semi-supervised vs Unsupervised

- Unsupervised: No labels for the target data

Source data

Target data

Fully-labeled

Single vs Multiple Source Domains

Source domain 1

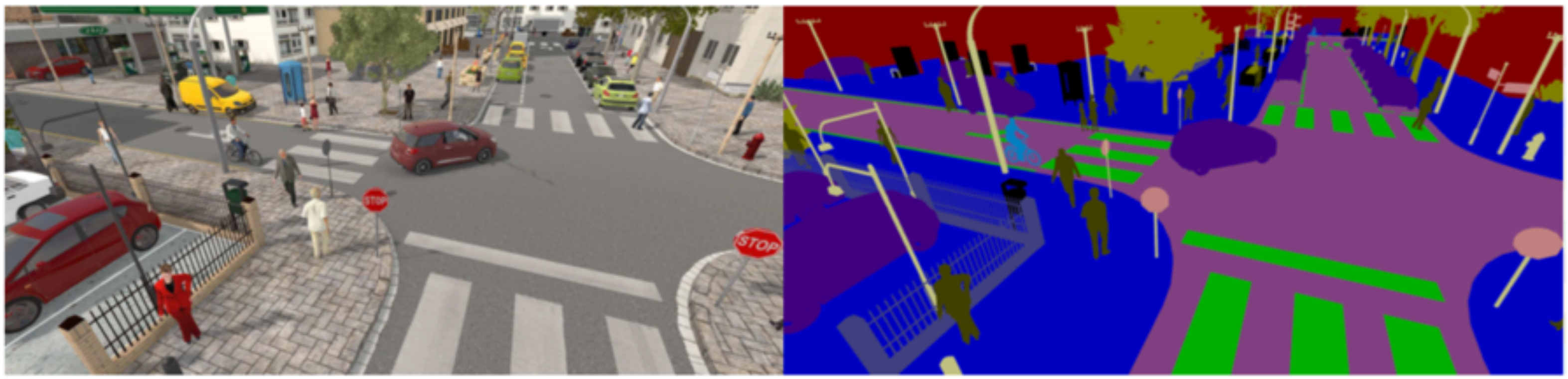
Source domain 2

Target domain

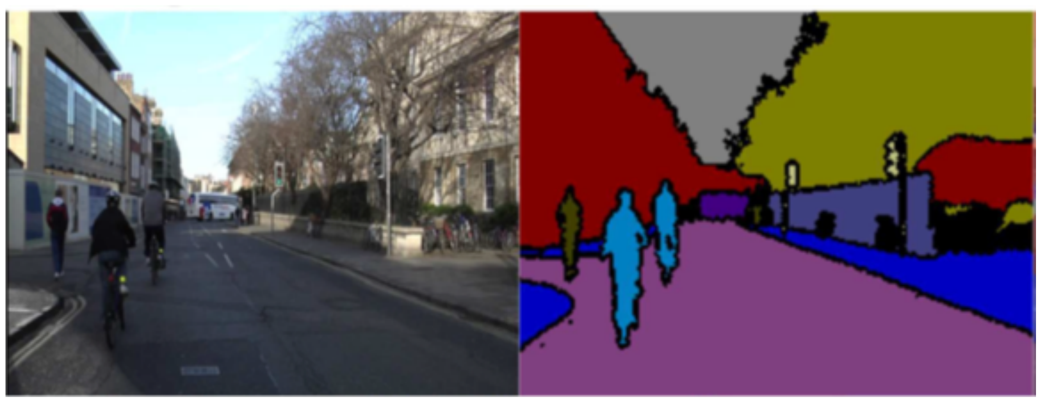
- Moving towards domain generalization

Domain Adaptation: Other Scenarios

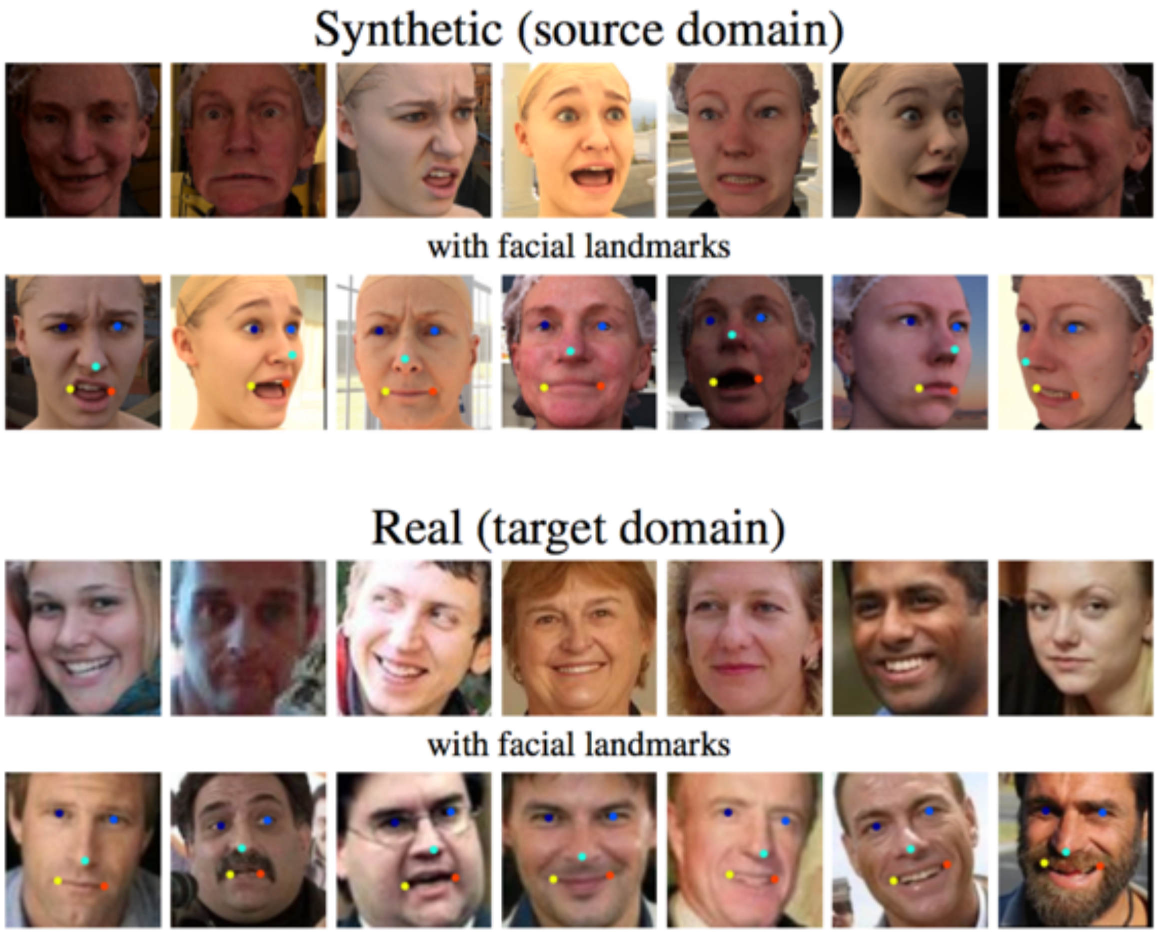
Synthetic (source domain)



Real (target domain)

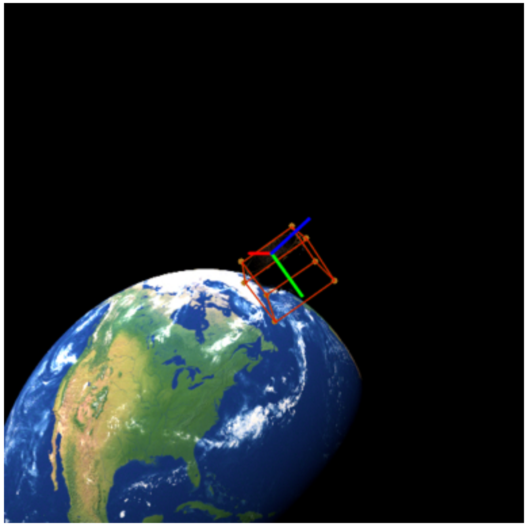


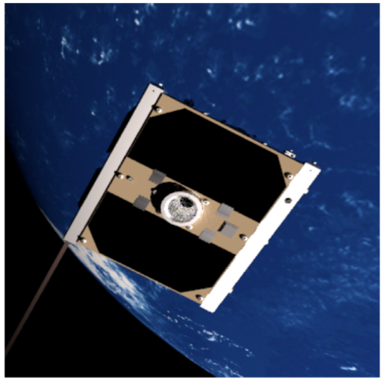
Domain Adaptation: Other Scenarios



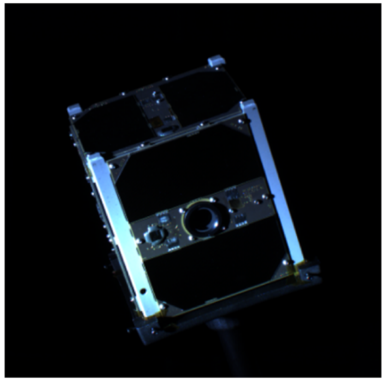
Domain Adaptation: Other Scenarios

Satellite 6D pose estimation





Synthetic (source)



Real (target)

Setup/Notation

- Each sample is represented by a feature vector:
  - In the traditional methods, e.g., bag of SURF features
  - More recently, features extracted by a deep backbone network

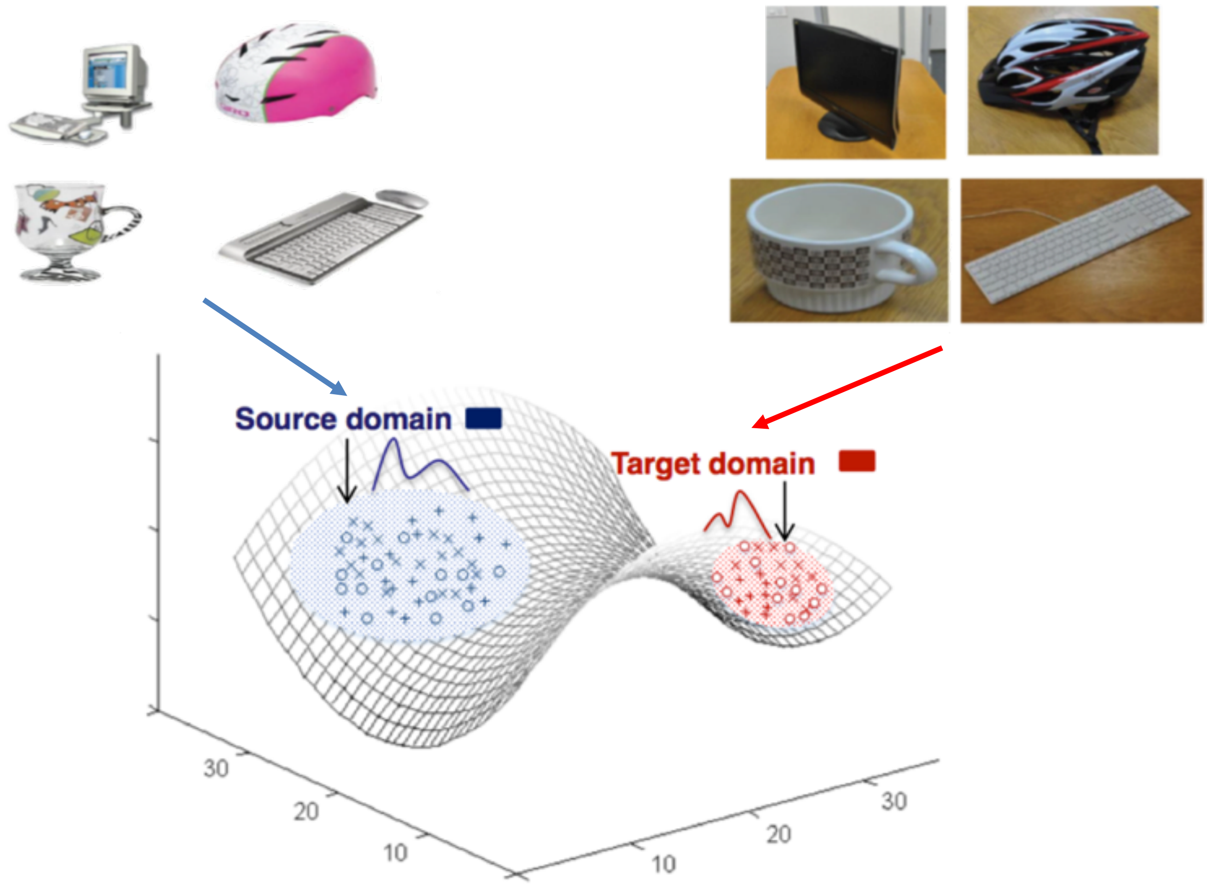
$\mathbf{X}_s = \{\mathbf{x}_s^i\}_{i=1}^n$

$\mathbf{X}_t = \{\mathbf{x}_t^j\}_{j=1}^m$

Label: $\{y_s^i\}_{i=1}^n$

Domain Shift

- The domain shift is defined as a difference in the distribution of the source and target samples



Domain Shift

- Typically, the literature focuses on the covariate shift case, where

$$p_t(x_t) \neq p_s(x_s)$$

- But

$$p_t(y|x_t) = p_s(y|x_s)$$

- The goal of domain adaptation is then often expressed as that of finding a transformation $T(.)$, such that

$$p_t(T(x_t)) = p_s(T(x_s))$$

Domain Shift

- Note that other types of shift have been studied. For example:
  - Long et al., ICCV 2013

$$p_t(y|x_t) \neq p_s(y|x_s) \quad \text{(concept shift)}$$

  - Gong et al., ICML 2016

$$p_t(y|T(x_t)) \neq p_s(y|T(x_s))$$

  - Kouw & Loog, 2018

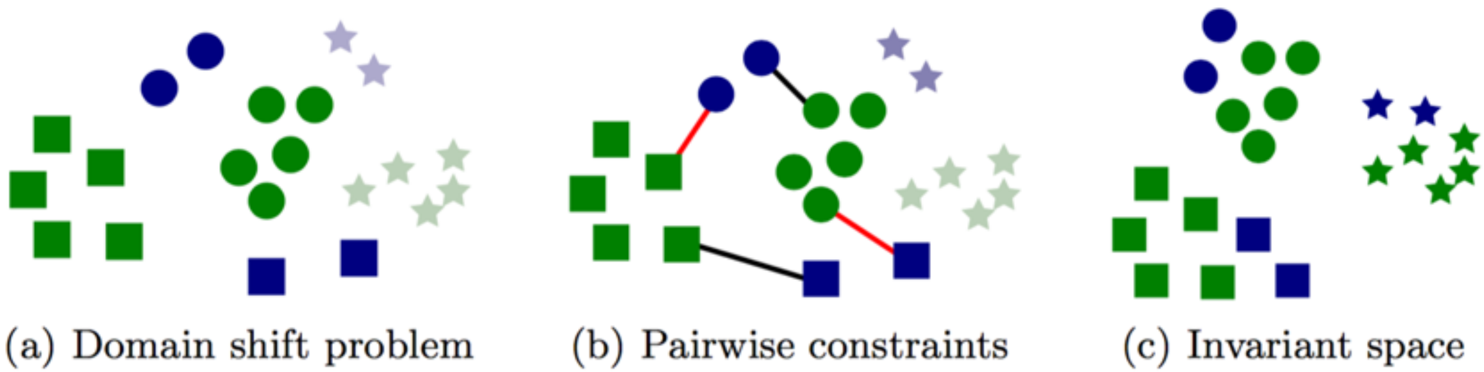
$$p_t(y) \neq p_s(y) \quad \text{(prior shift)}$$

- In this part, I will nonetheless focus on the covariate shift problem

Traditional Domain Adaptation

Metric Learning for Domain Adaptation

- Saenko et al., Adapting Visual Category Models to New Domains, ECCV 2010



- Learning a distance:

$$d_W(\mathbf{x}_s^i, \mathbf{x}_t^j) = (\mathbf{x}_s^i - \mathbf{x}_t^j)^T W (\mathbf{x}_s^i - \mathbf{x}_t^j)$$

Metric Learning for Domain Adaptation

- Semi-supervised domain adaptation: Pairwise constraints based on labels

$$
\begin{aligned}
d_W(\mathbf{x}_s^i, \mathbf{x}_t^j) &\leq u \text{ if } y^i = y^j \\
d_W(\mathbf{x}_s^i, \mathbf{x}_t^j) &\geq l \text{ if } y^i \neq y^j
\end{aligned}
$$

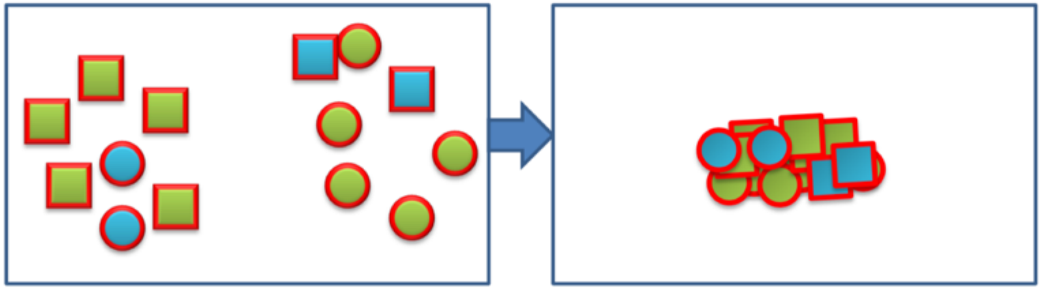
- Learning formulation:

$$
\begin{aligned}
\min_{W \succeq 0} \quad & \operatorname{tr}(W) - \log\det W \\
\text{s.t.} \quad & d_W(\mathbf{x}_s^i, \mathbf{x}_t^j) \leq u \text{ if } y^i = y^j \\
& d_W(\mathbf{x}_s^i, \mathbf{x}_t^j) \geq l \text{ if } y^i \neq y^j
\end{aligned}
$$

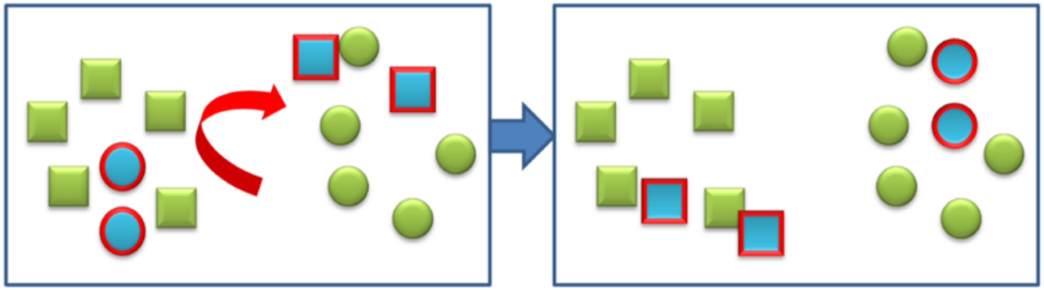
- Davis et al., ICML 2007:
  - Regularizer invariant to scaling and rotation
  - Efficient update based on a single constraint at a time

Metric Learning: Asymmetric Transformations

- The previous approach assumes:
  - Same feature dimensions for both domains
  - SPD matrix W
- This corresponds to a symmetric transformation



- Kulis et al., CVPR 2011 handles the asymmetric case



Metric Learning: Asymmetric Transformations

- Rely on similarity instead of distance:

$$\mathrm{sim}_W(\mathbf{x}_s^i, \mathbf{x}_t^j) = (\mathbf{x}_s^i)^T W \mathbf{x}_t^j$$

- The constraints can then be replaced with regularizers of the form:

$$(\max(0, l - (\mathbf{x}_s^i)^T W \mathbf{x}_t^j))^2$$ if the samples have the same label

$$(\max(0, (\mathbf{x}_s^i)^T W \mathbf{x}_t^j - u))^2$$ otherwise

- Replace the logdet regularizer with $\frac{1}{2}\|W\|_F^2$

- The formulation can be kernelized

From Semi-supervised to Unsupervised DA

- The previous approaches require some labeled target samples
- The unsupervised scenario assumes no target labels are available

Source data

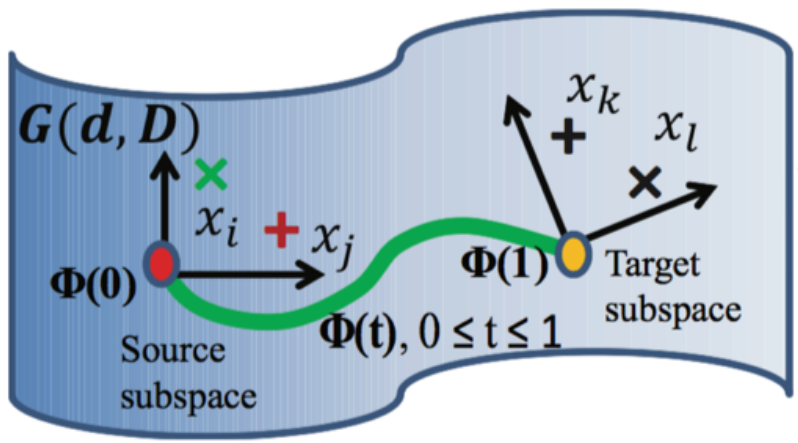
Fully-labeled

Target data

~~A few labels~~

Subspace Representations

- To model the data in each domain, several works have proposed to rely on subspace representations
  - This allows one to consider the entire data in one domain as a single entity

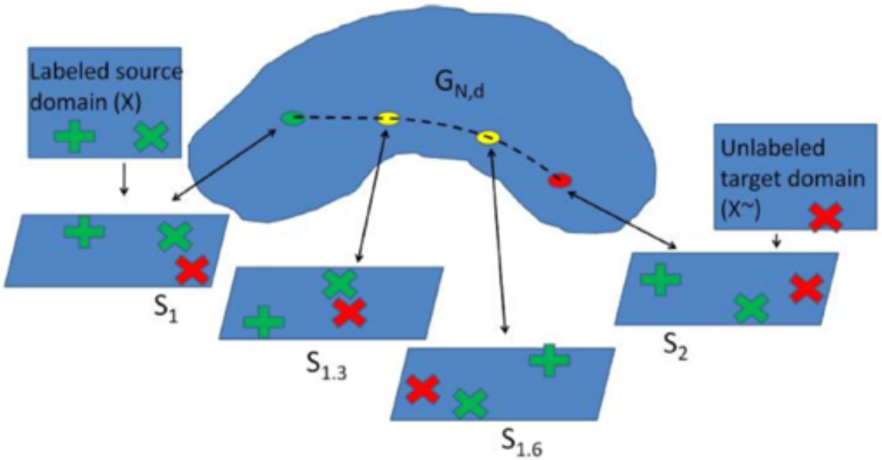


- Subspaces lie on Grassmann manifolds
  - Notions of Riemannian geometry, such as geodesics can be exploited

Subspace Representations

- Gopalan et al., ICCV 2011

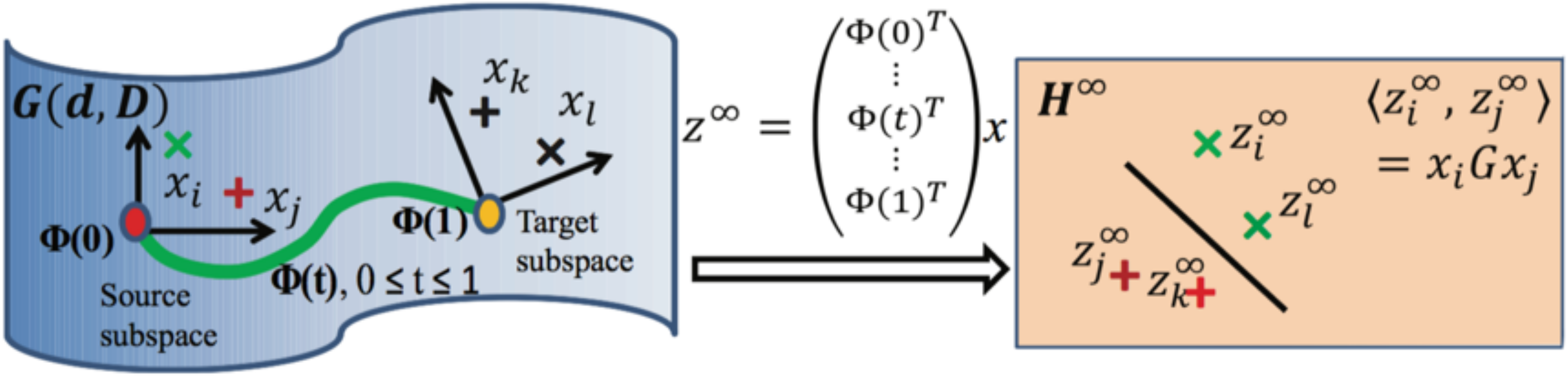
- Generating intermediate subspaces
  - Samples along the geodesic between the source and target subspaces



- Recognition
  - Project source and target samples on all subspaces
  - PLS on the resulting vector representation

Geodesic Flow Kernel

- Gong et al., CVPR 2012



- Instead of sampling, integrate over all subspaces
  - Projection over all subspaces generates infinite dimensional vector representations

- Inner product between two such vectors

$$\langle \boldsymbol{z}_i^\infty, \boldsymbol{z}_j^\infty \rangle = \int_0^1 (\boldsymbol{\Phi}(t)^{\mathrm{T}} \boldsymbol{x}_i)^{\mathrm{T}} (\boldsymbol{\Phi}(t)^{\mathrm{T}} \boldsymbol{x}_j)\, dt = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{G} \boldsymbol{x}_j$$

Subspace Alignment

- Fernando et al., ICCV 2013

- Don’t consider intermediate subspaces, align the source and target ones

- Solve

$$\mathbf{M}^* = \underset{\mathbf{M}}{\operatorname{argmin}} \|\mathbf{S}_s\mathbf{M} - \mathbf{S}_t\|_F^2$$

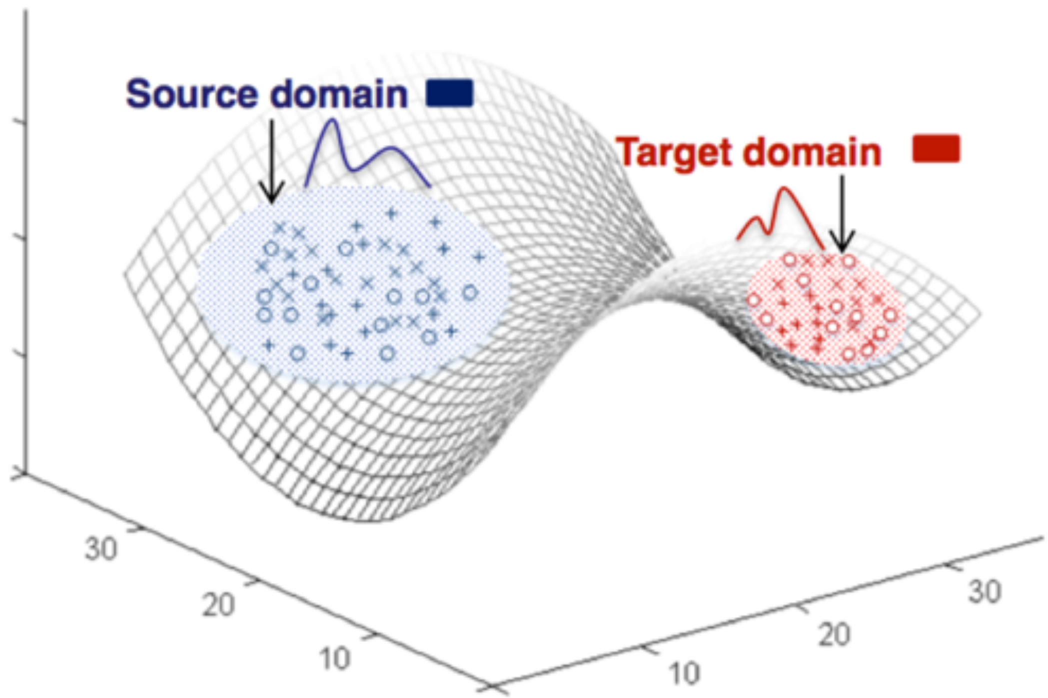
- Closed-form solution

$$\mathbf{M}^* = \mathbf{S}_s^T\mathbf{S}_t$$

Unsupervised DA: Matching Distributions

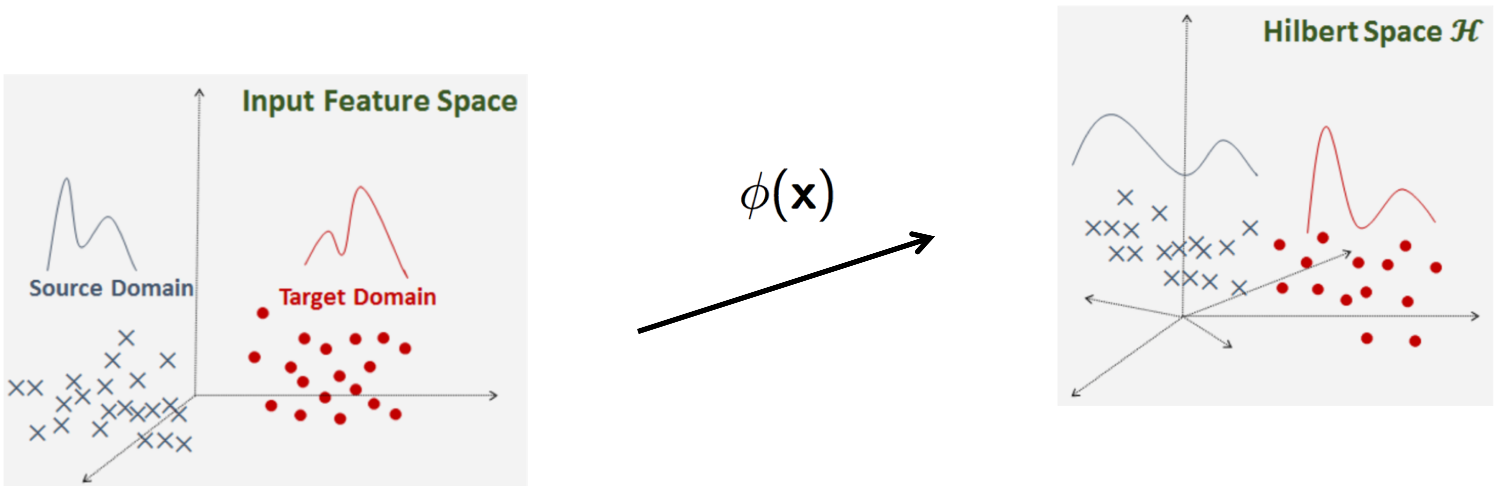
- While effective, subspace-based methods indirectly address the domain shift
  - Recall that it results from a distribution mismatch

- A popular DA approach therefore consists of aligning the distributions



Maximum Mean Discrepancy

- Compare the mean of two samples in Hilbert space
  - Gretton et al., JMLR 2012

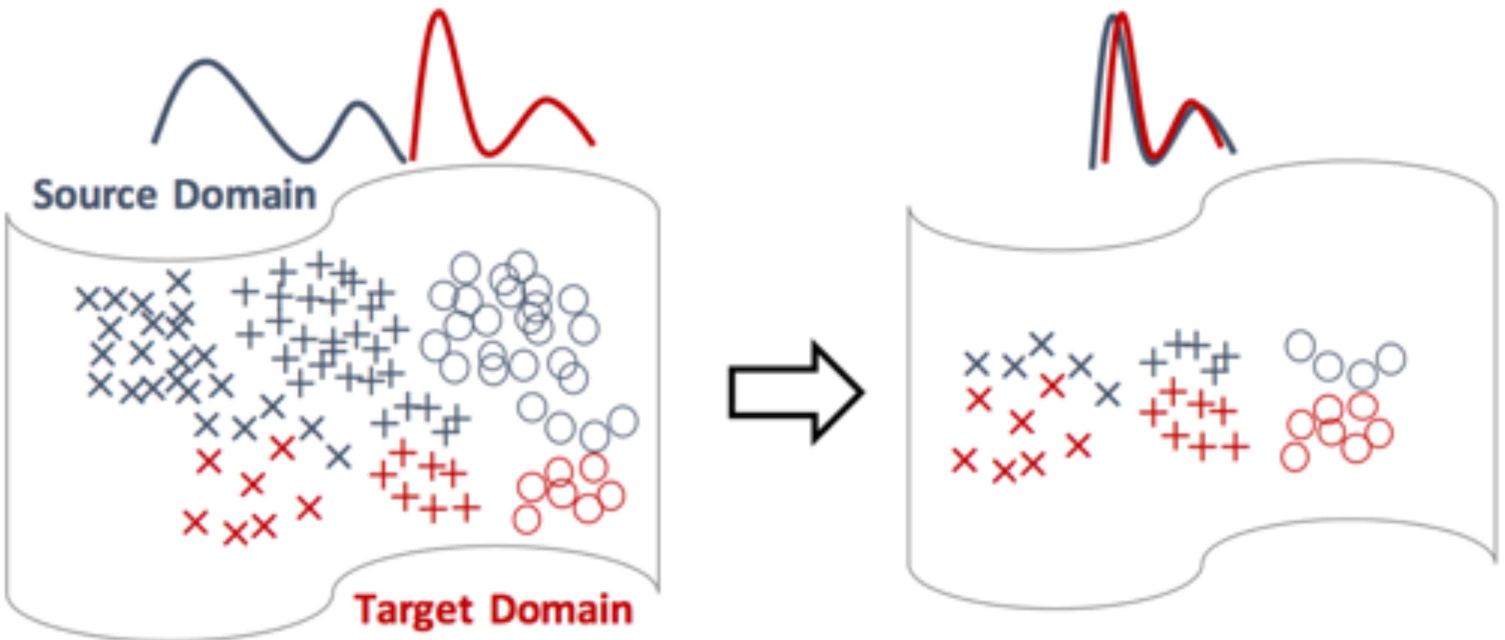


$$D_{MMD}(X_s, X_t) = \left\| \frac{1}{n}\sum_{i=1}^{n}\phi(x_s^i) - \frac{1}{m}\sum_{j=1}^{m}\phi(x_t^j) \right\|_{\mathscr{H}}$$

$$= \left(\sum_{i,j=1}^{n} \frac{k(x_s^i, x_s^j)}{n^2} + \sum_{i,j=1}^{m} \frac{k(x_t^i, x_t^j)}{m^2} - 2\sum_{i,j=1}^{n,m} \frac{k(x_s^i, x_t^j)}{nm} \right)^{\frac{1}{2}}$$

Sample Reweighting/Selection

- Assign a weight to each source sample to make the distributions similar



Sample Reweighting/Selection

- Gretton et al., JRSS 2012: Sample reweighting

$$\min_{\beta} \quad \left\| \frac{1}{n} \sum_{i=1}^{n} \beta_i \phi(\mathbf{x}_s^i) - \frac{1}{m} \sum_{i=1}^{m} \phi(\mathbf{x}_t^i) \right\|^2$$

MMD

$$\text{s.t.} \quad \beta_i \in [0, B] \, , \forall\, 1 \leq i \leq n$$

Bound on the weights

$$\left| \sum_{i=1}^{n} \beta_i - n \right| \leq n\epsilon$$

Encourage the weights to define a probability distribution

Sample Reweighting/Selection

- Gong et al., ICML 2013: Sample selection

$$\min_{\boldsymbol{\alpha}} \quad \left\| \frac{1}{\sum_{i=1}^{n} \alpha_i} \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_s^i) - \frac{1}{m} \sum_{i=1}^{m} \phi(\mathbf{x}_t^i) \right\|^2$$

MMD

$$\text{s.t.} \quad \alpha_i \in \{0, 1\}, \ \forall\, 1 \leq i \leq n$$

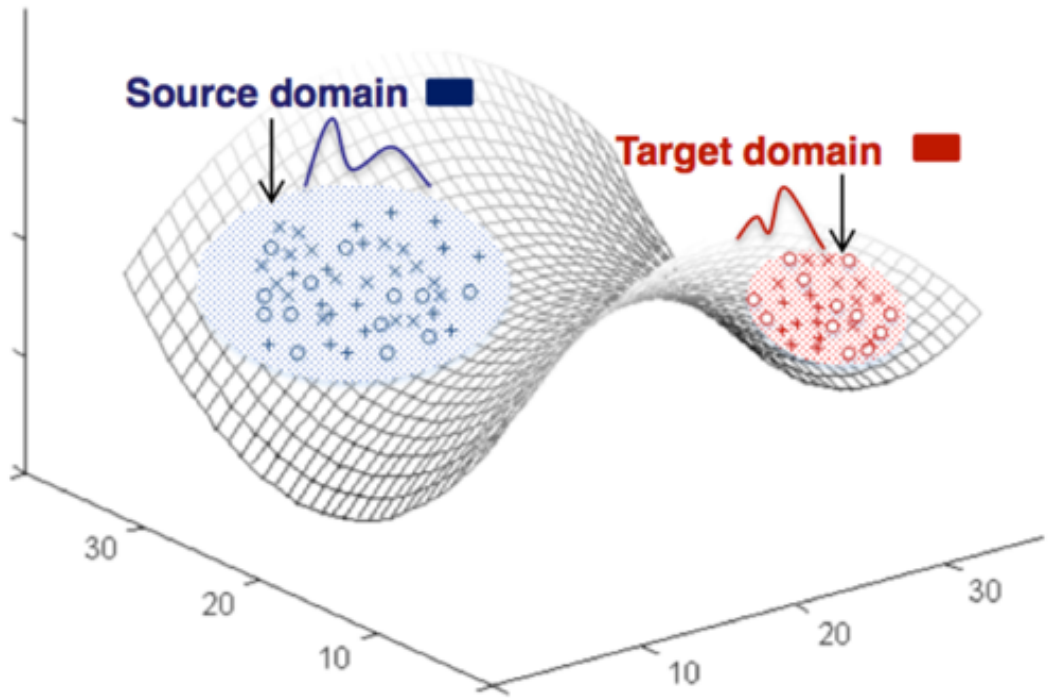
Binary weights

$$\frac{1}{\sum_{i=1}^{n} \alpha_i} \sum_{i=1}^{n} \alpha_i y_c^i = \frac{1}{n} \sum_{i=1}^{n} y_c^i, \ \forall 1 \leq c \leq C$$

Keep the same proportion of sample in each class

Sample Reweighting/Selection

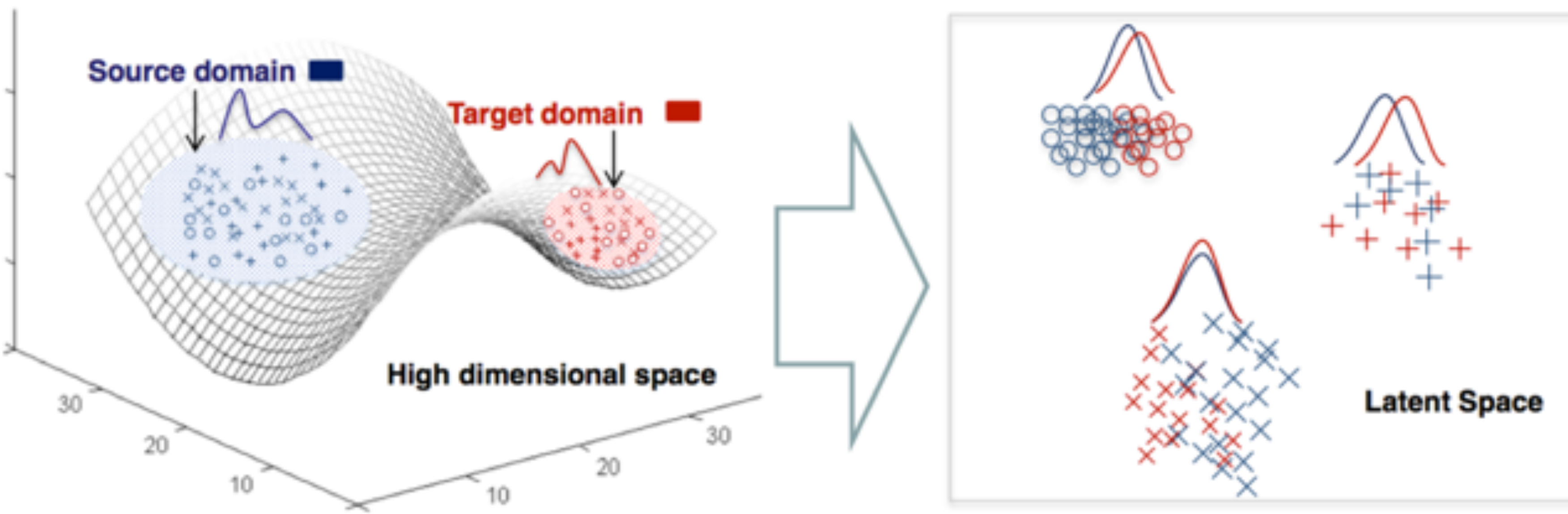
- What happens if the original distributions are very different?



- Selecting/reweighting samples will not be sufficient to align the distributions

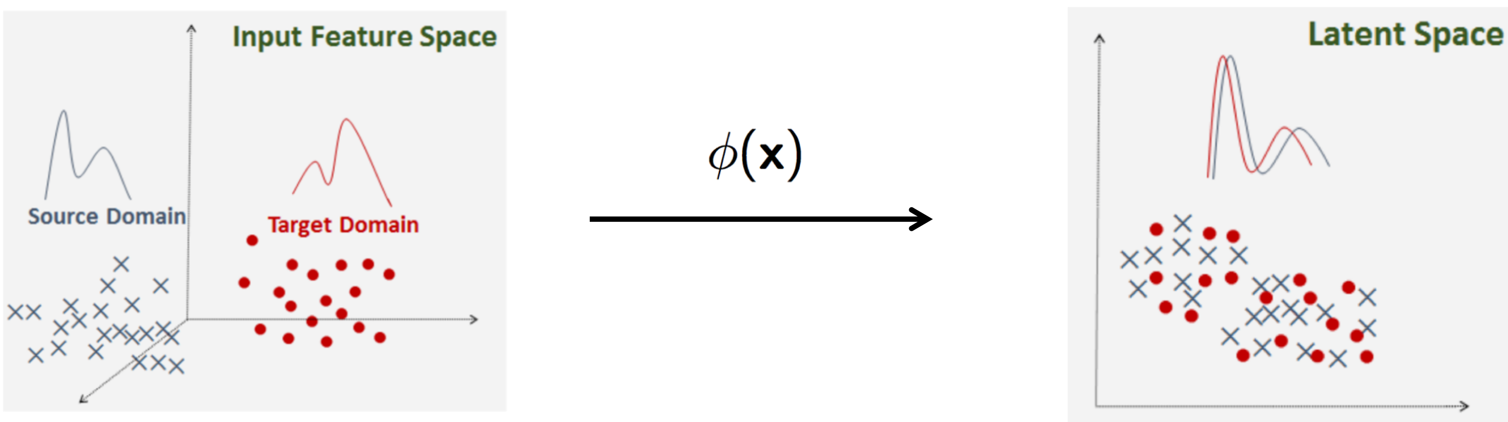
Transformation Learning

- Learn a mapping to a latent space where the distributions are similar



Transfer Component Analysis (TCA)

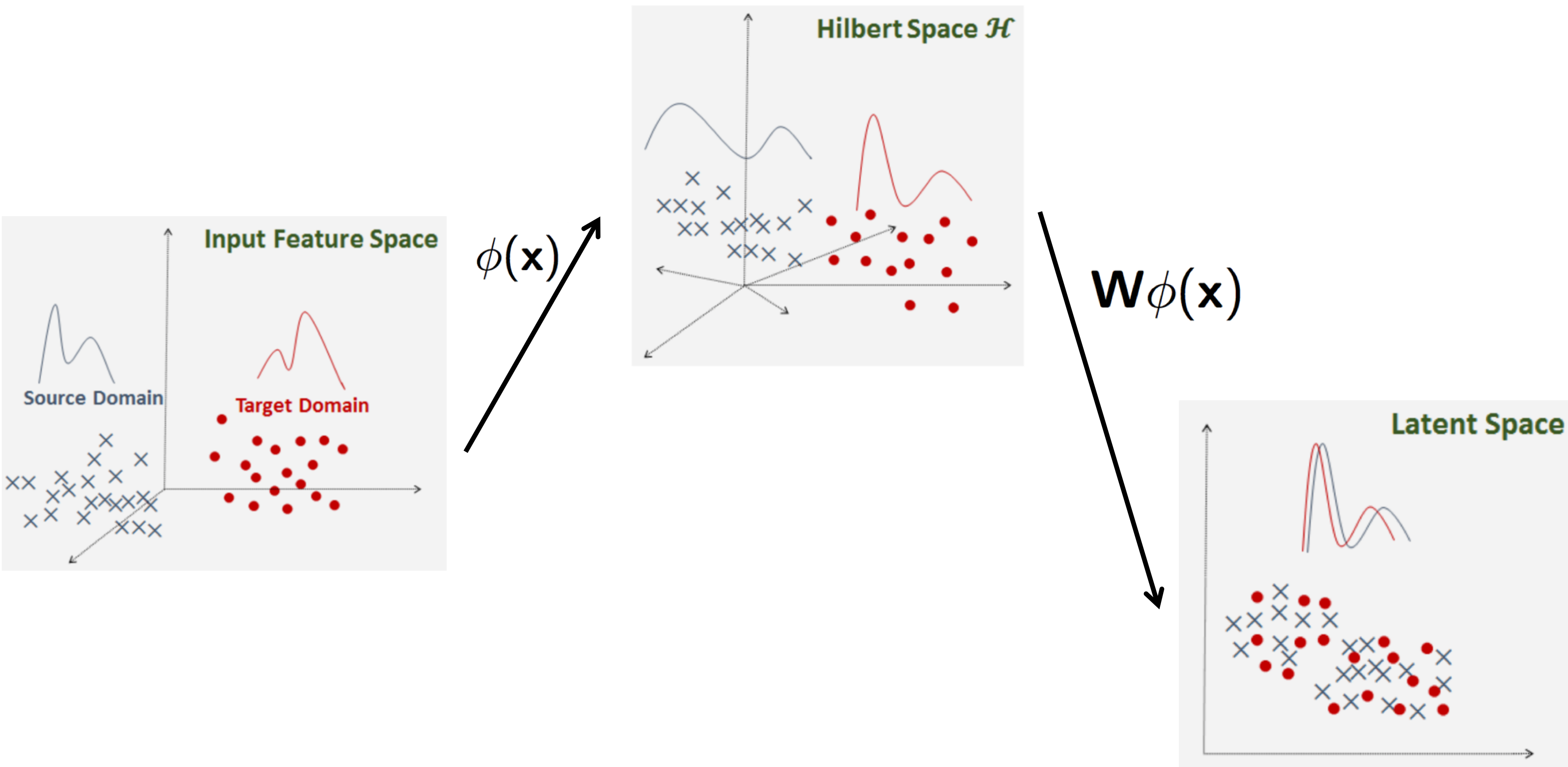
- Pan et al., TNN 2011
    - Motivation: Learn a nonlinear mapping that minimizes the MMD



- Would involve learning a kernel matrix, which is ill-constrained

TCA: Simplification

- Relies on a projection of the empirical kernel map to a latent space



TCA: Simplification

- Yields a new kernel matrix $\tilde{K} = KWW^T K$

- The MMD becomes

$$D^2_{MMD}(X_s, X_t) = Tr((KWW^T K)L)$$

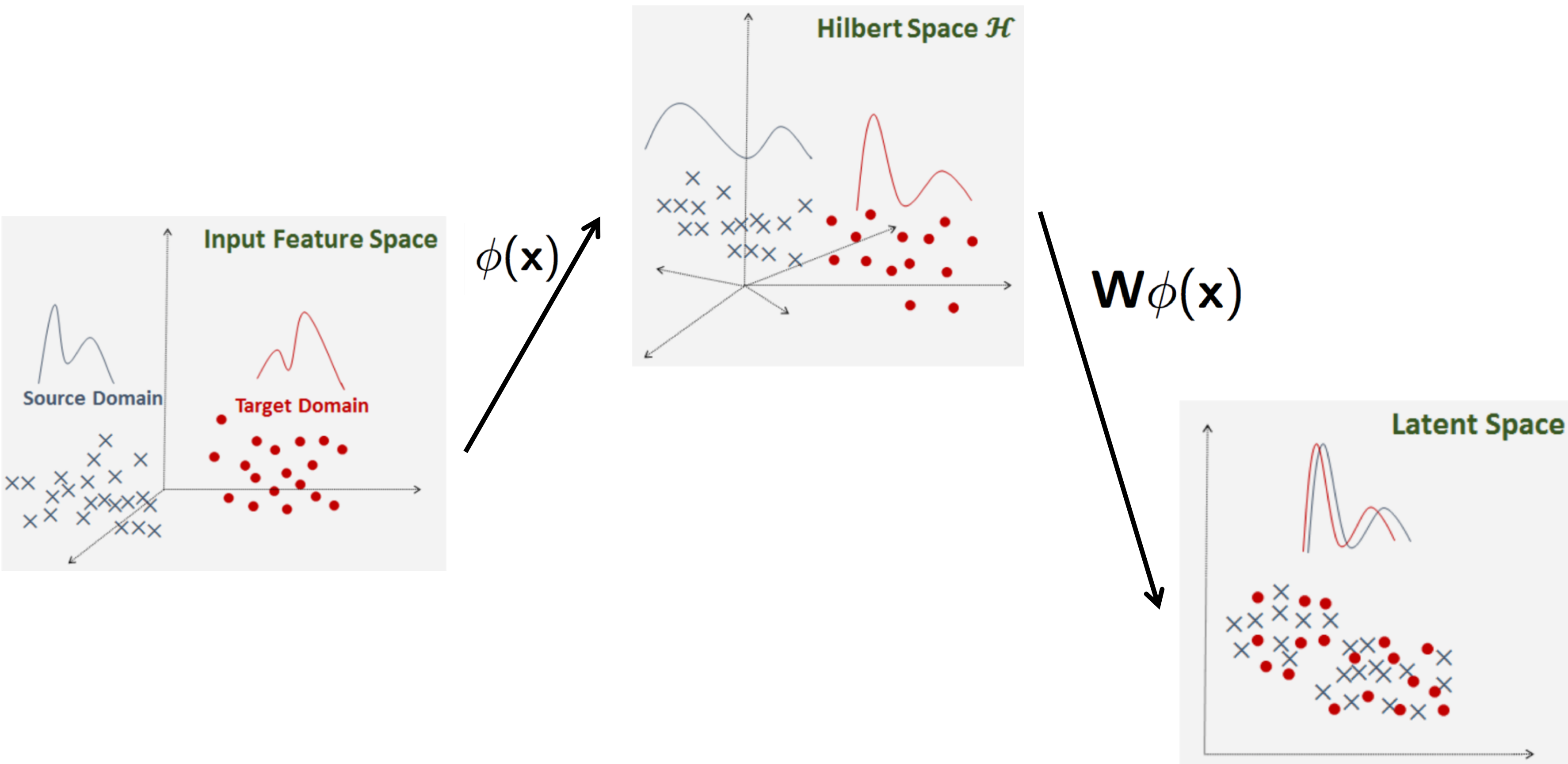
- To better constrain the problem, regularize the data variance $\tilde{\Sigma}$

- Formulation

$$\min_W \; Tr(W^T KLKW) + \mu Tr(W^T W) \;\; \text{s.t.} \;\; \tilde{\Sigma} = I_d$$

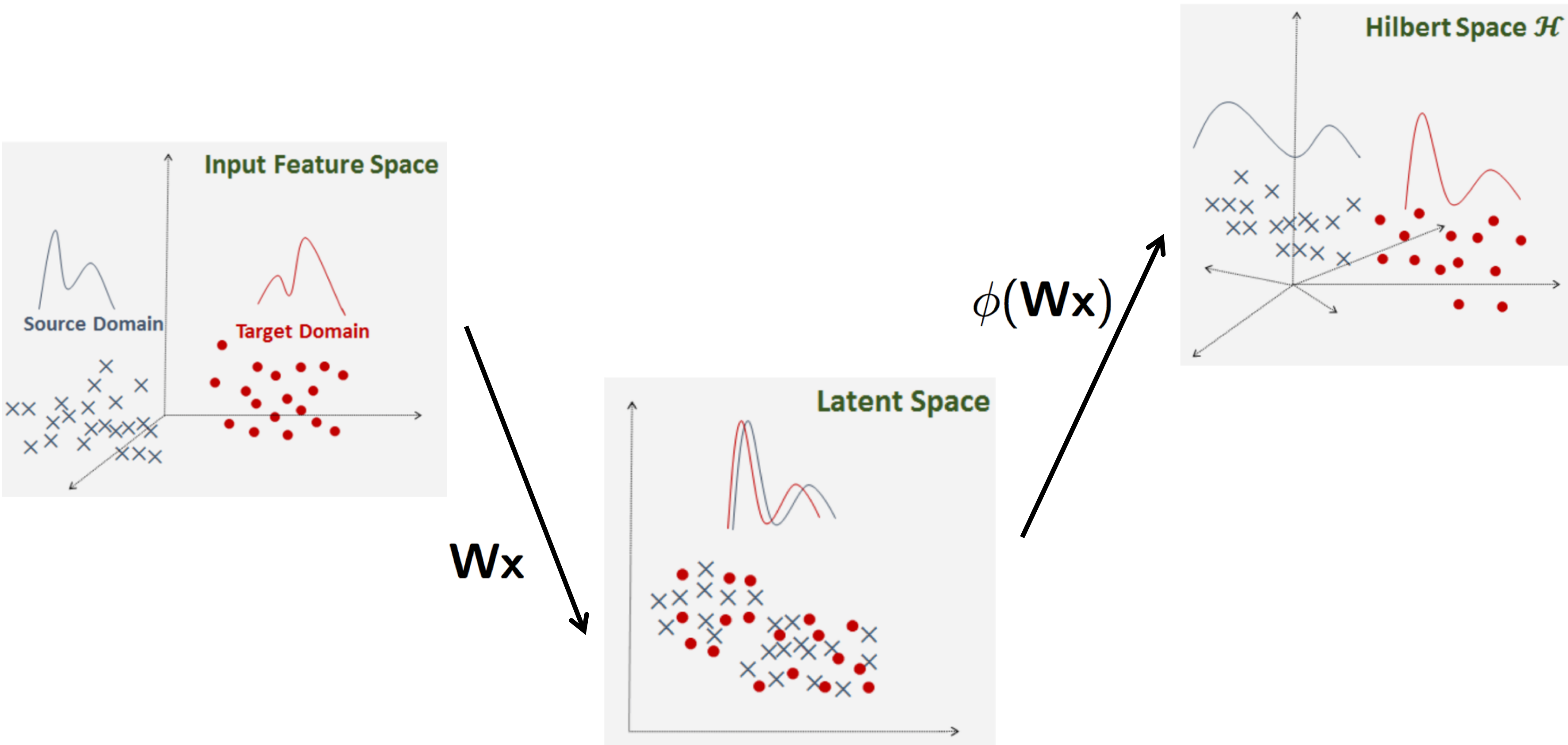
TCA Interpretation

- TCA compares the mean of each domain after projection to the latent space
- MMD not truly computed in Hilbert space



Domain Invariant Projection (DIP)

- Baktashmotlagh et al., ICCV 2013
  - Learn a latent representation such that the MMD is minimized
  - MMD truly makes use of the Hilbert space



Domain Invariant Projection (DIP)

- MMD: $D_{MMD}(W^TX_s, W^TX_t) = \left\| \frac{1}{n}\sum_{i=1}^{n} \phi(W^Tx_s^i) - \frac{1}{m}\sum_{j=1}^{m} \phi(W^Tx_t^j) \right\|_{\mathscr{H}}$

- With a Gaussian kernel

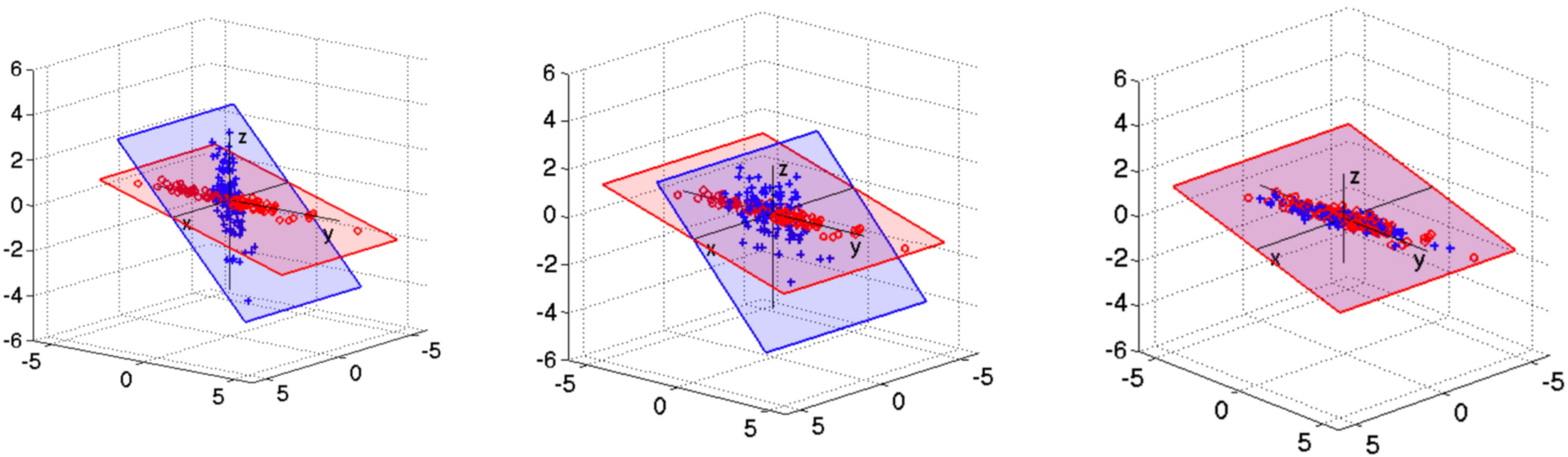
$$
\begin{aligned}
&D^2_{MMD}(W^TX_s, W^TX_t) = \\
&\frac{1}{n^2}\sum_{i,j=1}^{n} \exp\left(-\frac{(x_s^i - x_s^j)^T W W^T (x_s^i - x_s^j)}{\sigma}\right) \\
&+\frac{1}{m^2}\sum_{i,j=1}^{m} \exp\left(-\frac{(x_t^i - x_t^j)^T W W^T (x_t^i - x_t^j)}{\sigma}\right) \\
&-\frac{2}{mn}\sum_{i,j=1}^{n,m} \exp\left(-\frac{(x_s^i - x_t^j)^T W W^T (x_s^i - x_t^j)}{\sigma}\right)
\end{aligned}
$$

- Formulation

$$
\begin{aligned}
W^* &= \underset{W}{\operatorname{argmin}} \; D^2_{MMD}(W^TX_s, W^TX_t) \\
&\text{s.t.} \quad W^TW = I_d \,,
\end{aligned}
$$

Comparing Covariances

- Sun et al., AAAI 2016: CORAL
  - First de-correlate the source features
  - Then re-correlate them with the target correlation



- Mathematically: $\min_{A} \| A^{\top} C_S A - C_T \|_F^2$

- Note that, as opposed to the means in the MMD, the covariance matrices are computed in the original space

Other Distribution Distances

- f-divergences:

$$D_f(s||t) = \int f\left(\frac{s(x)}{t(x)}\right) t(x) dx$$

- In practice, the distributions can be estimated using KDE

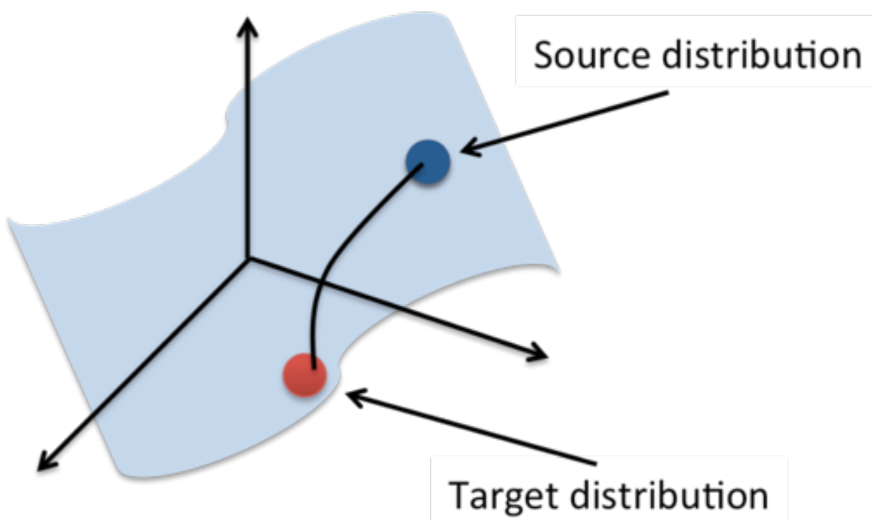
- In particular, the KL-divergence:

$$KL(s||t) = \int s(x) \log \frac{s(x)}{t(x)} dx$$

Hellinger Distance

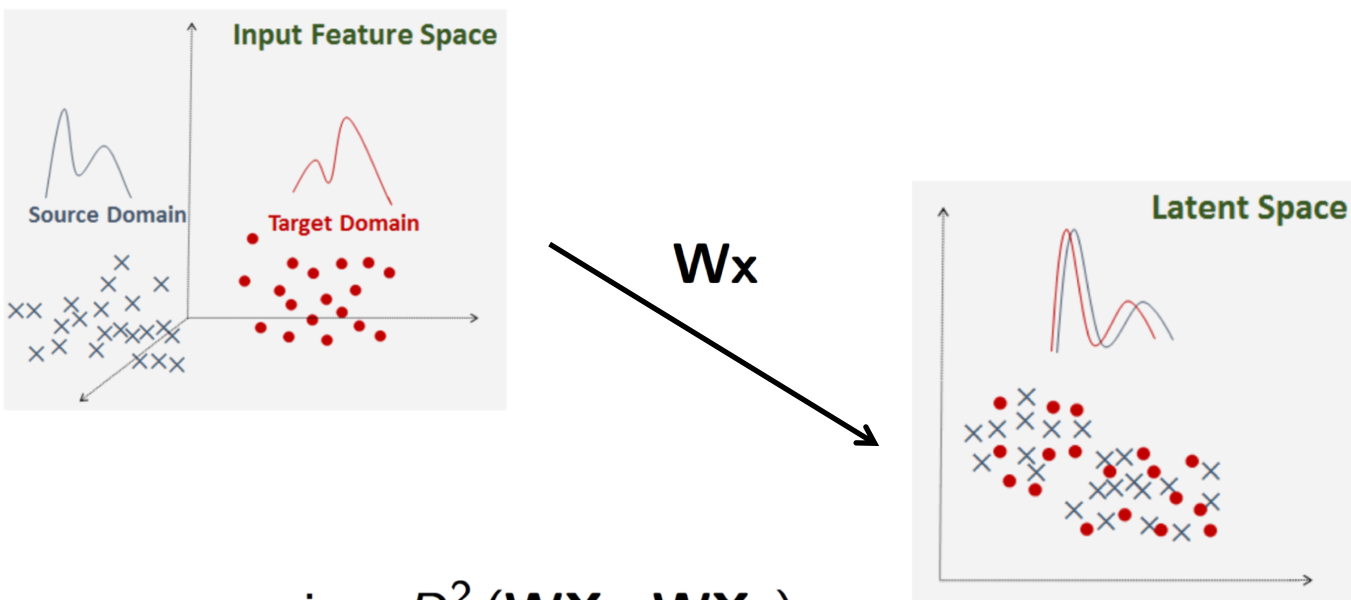
$$D_H^2(s\|t) = \int \left(\sqrt{s(x)} - \sqrt{t(x)} \right)^2 dx$$

- Related to the geodesic distance on the statistical manifold
  - The length of a curve is the same under both distances



Statistically Invariant Embedding (SIE)

- Baktashmotlagh et al., CVPR 2014
  - Hellinger distance instead of MMD
  - Applied to sample selection and transformation learning



$$\begin{aligned} \min_{\mathbf{W}} \quad & D_H^2(\mathbf{W}\mathbf{X}_s, \mathbf{W}\mathbf{X}_t) \\ \text{s.t.} \quad & \mathbf{W}\mathbf{W}^T = \boldsymbol{I} \end{aligned}$$

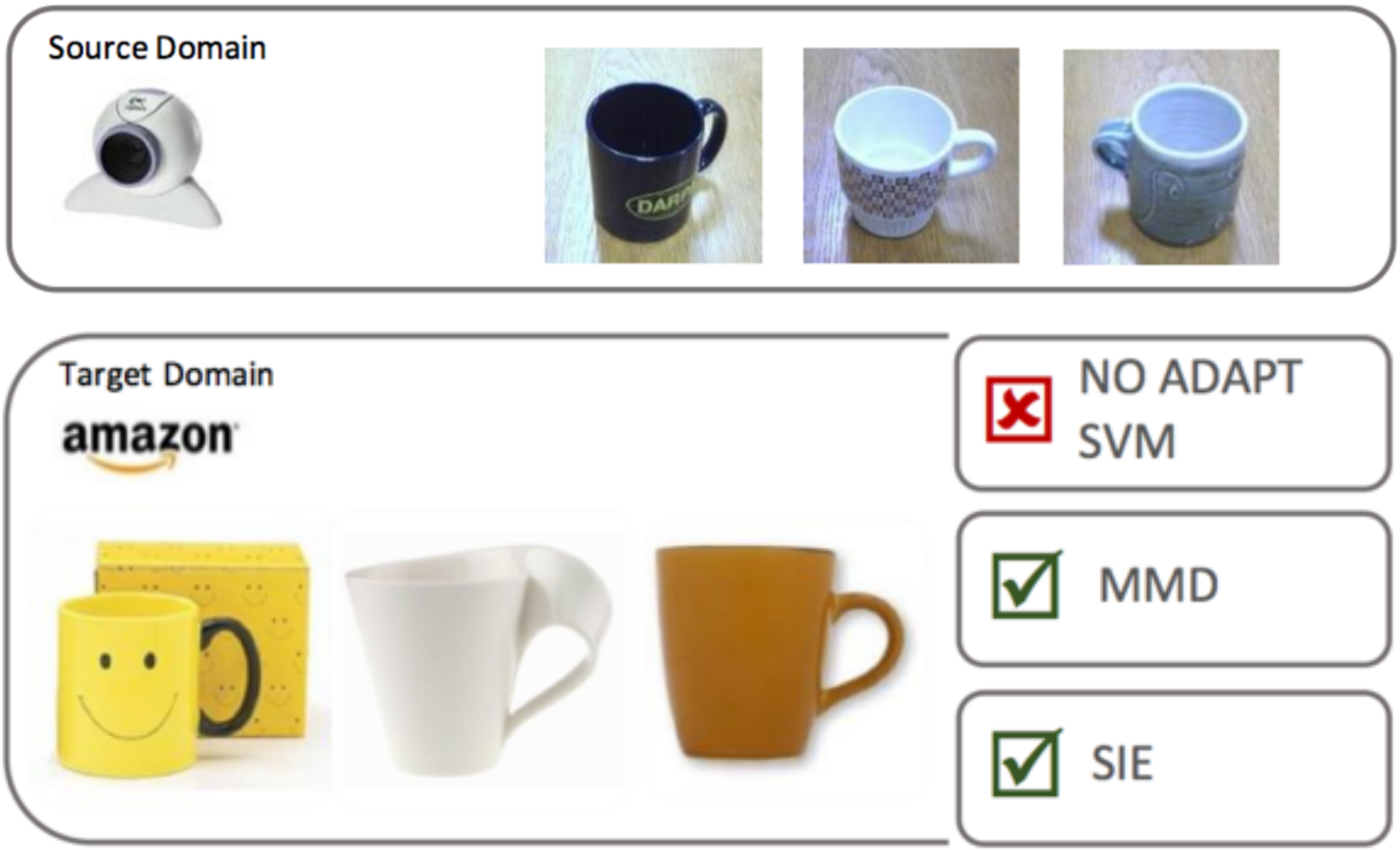
Empirical Evaluation: Dataset

- Introduced by Saenko et al., ECCV 2010
- Complemented with Caltech by Gong et al., CVPR 2012
- 4 domains, 10 classes
- BoW of SURF features
- Decaf features

Empirical Evaluation: Results (SURF)

Method	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$	Avg.
NO ADAPT-SVM	33.6 ± 1.7	31.1 ± 0.9	75.2 ± 2.6	36.9 ± 1.2	33.4 ± 1.1	80.2 ± 2.5	44
SVMA (Duan et al., 2012)	33.43 ± 1.24	31.40 ± 0.87	74.44 ± 2.21	36.63 ± 1.08	33.52 ± 0.77	74.97 ± 2.65	41.1
DAM (Duan et al., 2012)	33.50 ± 1.29	31.52 ± 0.88	74.68 ± 2.14	34.73 ± 1.14	31.18 ± 1.25	68.34 ± 3.16	40.2
GFK (Gong et al., 2012)	37.7 ± 1.8	33.3 ± 1.3	79.9 ± 2.8	41.5 ± 1.8	34.5 ± 0.9	76.7 ± 1.4	44.8
TCA (Pan et al., 2011)	39.6 ± 1.2	34 ± 1.1	80.4 ± 2.6	40.2 ± 1.1	33.7 ± 1.1	77.5 ± 2.5	42.8
SA (Fernando et al., 2013)	41.1 ± 1.6	35.4 ± 1.8	84.4 ± 2.4	38.2 ± 1.4	33.3 ± 1.2	83.3 ± 1.6	48.7
KMM (Huang et al., 2006)	38 ± 1.8	34.3 ± 1.2	82.0 ± 1.7	39.0 ± 1.2	35.3 ± 1.0	86.8 ± 2.0	47.7
DME-MMD	40.5 ± 1	39 ± 0.5	86.7 ± 1.2	42.5 ± 1.5	37 ± 0.9	86.4 ± 1.8	50.9
DME-MMD (Poly)	40.8 ± 0.9	39.1 ± 0.6	87.1 ± 1.0	41.3 ± 1.3	36.8 ± 0.9	85.8 ± 2.2	50.4
DME-H	39.1 ± 0.6	38.9 ± 0.4	88.6 ± 1.0	44.1 ± 0.8	39.9 ± 0.7	89.3 ± 0.5	52.3

Empirical Evaluation: Results



Empirical Evaluation: Results

Empirical Evaluation: Results

Source Domain

Target Domain

amazon

NO ADAPT SVM

MMD

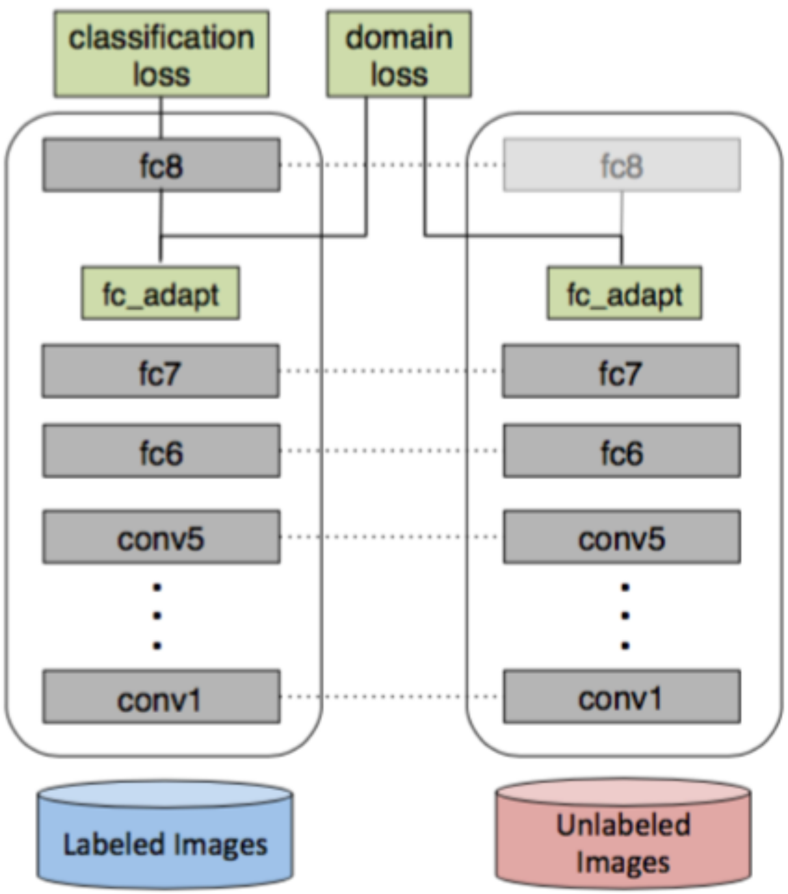
SIE

Empirical Evaluation: Results (Decaf)

Method	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$	Avg.
NO ADAPT-SVM	79.2 ± 2.3	73.4 ± 2.0	95.6 ± 1.1	75.3 ± 1.5	69.5 ± 1.1	99.4 ± 0.6	81.9
SVMA (Duan et al., 2012)	85.37	78.14	96.71	74.36	70.58	96.6	82.7
DAM (Duan et al., 2012)	87.88	81.27	96.31	76.6	74.32	93.8	84.2
GFK (Gong et al., 2012)	84.2 ± 2.3	77.5 ± 2.0	96.4 ± 1.1	85.4 ± 1.7	77.1 ± 0.5	99.5 ± 0.3	86.8
TCA (Pan et al., 2011)	84.1 ± 1.6	77.7 ± 1.9	95.9 ± 0.8	83.8 ± 1.0	76.5 ± 0.9	98.6 ± 0.9	85.6
SA (Fernando et al., 2013)	90.1 ± 0.9	83.9 ± 1.6	96.8 ± 1.6	85.0 ± 3.3	78.7 ± 2.8	99.3 ± 0.7	86.5
KMM (Huang et al., 2006)	84.3 ± 2.4	77.4 ± 1.1	96.2 ± 1.8	75.5 ± 3.2	72.8 ± 1.9	97.9 ± 0.9	83.6
DME-MMD	82.9 ± 2.9	77.5 ± 2.7	96.4 ± 1.2	82.1 ± 1.9	78.6 ± 1.4	98.8 ± 0.3	86.2
DME-H	84.5 ± 2.5	79.6 ± 1.8	97 ± 0.9	83.9 ± 1.1	77.9 ± 1.4	99.7 ± 0.4	86.7

MMD-based Network

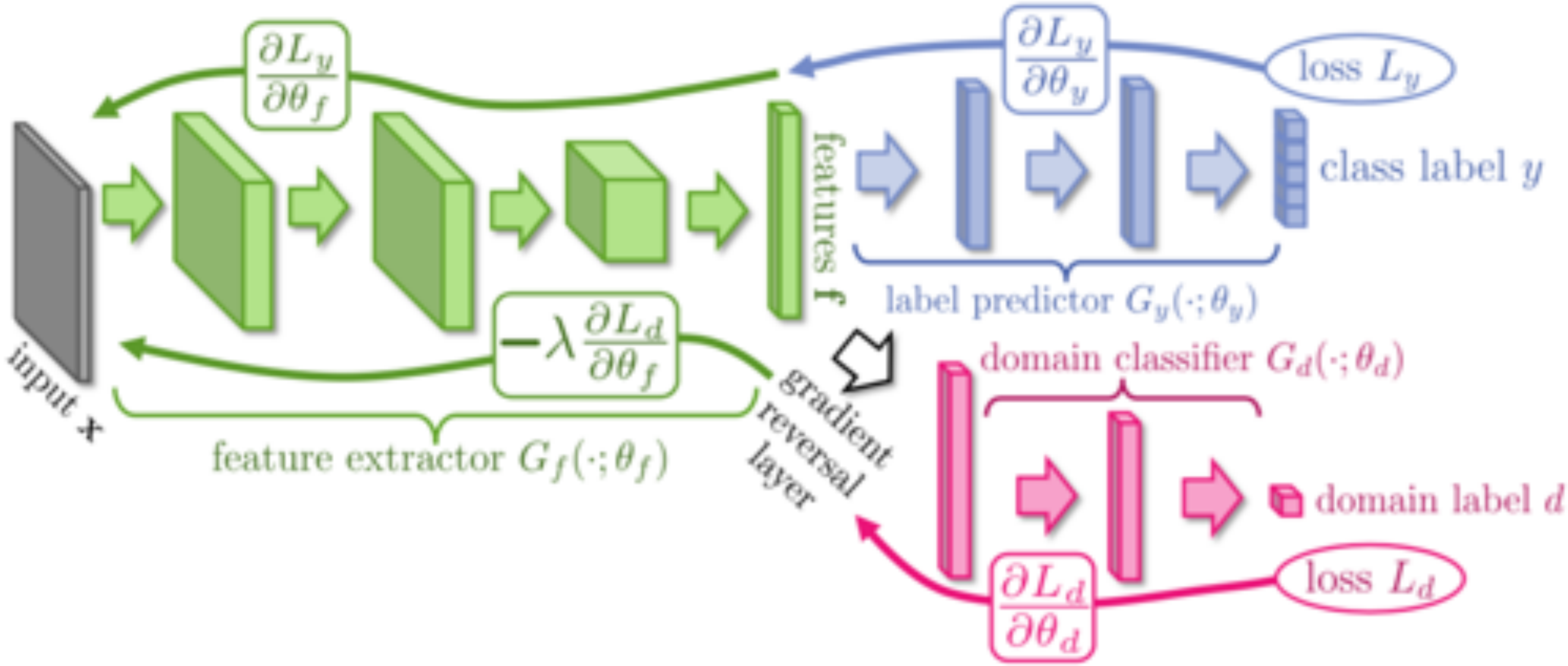
- Deep Domain Confusion: Tzeng et al., 2014



$$\mathcal{L} = \mathcal{L}_C(X_L, y) + \lambda \mathrm{MMD}^2(X_S, X_T)$$

Domain Adversarial Networks

- Ganin & Lempitsky, ICML 2015; Ajakan et al., 2014
  - With domain-invariant features, classifying from which domain a sample comes should be difficult



- Shown to optimize a H-divergence between the source and target data

Deep Learning for Domain Adaptation (Office 31)

	Method	$A \rightarrow D$	$D \rightarrow W$	$W \rightarrow D$
Deep	DAN (Ganin)	67.3	94.0	93.7
	DDC (Tzeng)	59.4	92.5	91.7
Shallow	DIP	53.2	86.3	93.7
	SIE	51.6	87.4	92.9

Summary

- Learning transformations to match distributions
  - Well-motivated and intuitive
  - Effective in practice

- Subspace-based representations are also powerful
  - Subspace alignment is simple and effective

- End-to-end learning has surpassed the traditional approach
  - Many ideas used in the past can be and have been translated to deep networks