An Introduction to Transfer Learning and Domain Adaptation

Amaury Habrard

Laboratoire Hubert Curien, UMR CNRS 5516, Université de Saint-Etienne

EPAT 2014

Some resources

- List of transfer learning papers
  http://www1.i2r.a-star.edu.sg/~jspan/conferenceTL.htm
- List of available softwares
  http://www.cse.ust.hk/TL/index.html
- Surveys
    - Patel, Gopalan, Chellappa. Visual Domain Adaptation: An Overview of Recent Advances. Tech report, 2014.
    - Qi Li. Literature Survey: Domain Adaptation Algorithms for Natural Language Processing, Tech report, 2012
    - Margolis. A Literature Review of Domain Adaptation with Unlabeled Data. Tech report 2011.
    - Pan and Yang. A survey on Transfer Learning', TKDE 2010.
    - J. Quionero-Candela and M. Sugiyama and A. Schwaighofer and N.D. Lawrence. Dataset Shift in Machine Learning. MIT Press.

Credits and acknowledgments

Documents used for this talk:

- D. Xu, K. Saenko, I. Tsang. Tutorial on Domain Transfer Learning for Vision Applications, CVPR'12.
- S. Pan, Q. Yang and W. Fan. Tutorial: Transfer Learning with Applications, IJCAI'13.
- S. Ben-David. Towards Theoretical Understanding of Domain Adaptation Learning, workshop LNIID at ECML'09.
- F. Sha and B. Kingsbury. Domain Adaptation in Machine Learning and Speech Recognition, Tutorial - Interspeech 2012.
- K. Grauman. Adaptation for Objects and Attributes, workshop VisDA at ICCV'13
- J. Blitzer and H. DauméIII. Domain Adaptation, Tutorial - ICML 2010.
- A. Habrard, JP Peyrache, M. Sebban. Iterative Self-labeling Domain Adaptation for Linear Structured Image Classification, IJAIT 2013.
- A. Habrard, JP Peyrache, M. Sebban. Boosting for unsupervised domain adaptation, ECML 2013

Transfer Learning

Definition [Pan, TL-IJCAI'13 tutorial]

Ability of a system to recognize and apply knowledge and skills learned in previous domains/tasks to novel domains/tasks

An example

- We have **labeled** images from a **Web image** corpus
- Is there a `Person` in **unlabeled** images from a **Video** corpus ?

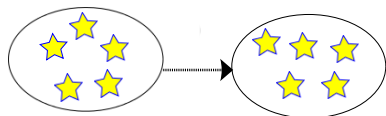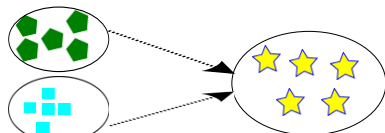Person   no Person   Is there a Person?

Outline

1. Introduction/Motivation

2. Reweighting/Instance based methods

3. Theoretical frameworks

4. Feature/projection based methods

5. Adjusting/Iterative methods

6. A quick word on model selection

Introduction

Training and test data are
from the same domain

Training and test data are
from different domains

- Domains are modeled as probability distributions over an instance space
- Tasks associated to a domain (classification, regression, clustering, ...)
- **Objective:** From source to target
  $\Rightarrow$ Improve a target predictive function in the target domain using knowledge from the source domain

A Taxonomy of Transfer Learning
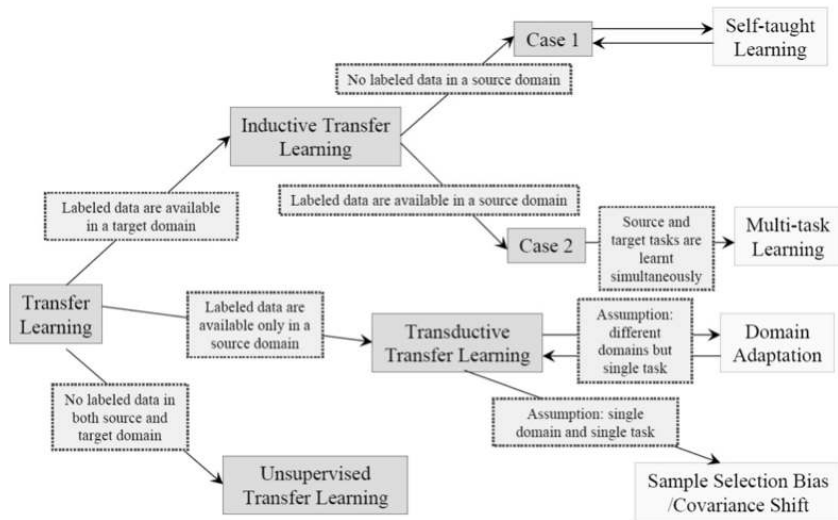
"A survey on Transfer Learning" [Pan and Yang, TKDE 2010]

In this presentation

- We will make a focus on domain adaptation
- We will focus on classification tasks

$\Rightarrow$ How can we learn, using labeled data from a source distribution, a low-error classifier for another related target distribution?
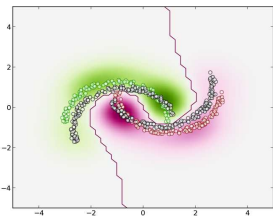
$\Rightarrow$ "Hot topic" - tutorials at ICML 2010, CVPR 2012, Interspeech 2012, workshops at ICCV 2013, NIPS 2013,ECML 2014
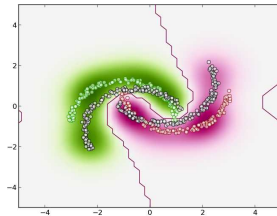
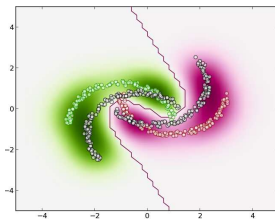$\Rightarrow$ Motivating examples
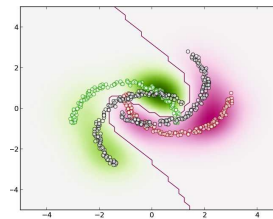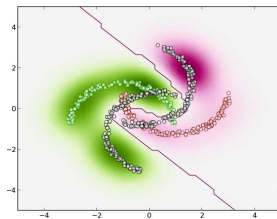
A toy problem: Inter-twinning moons
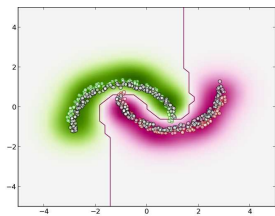
(a) 10°

(b) 20°

(c) 30°

(d) 40°

(e) 50°

(f) 70°

Intuition and motivation from a CV perspective

TRAIN ➡ TEST

- "Can we train classifiers with Flickr photos, as they have already been collected and annotated, and hope the classifiers still work well on mobile camera images?" [Gonq et al., CVPR 2012]

- "object classifiers optimized on benchmark dataset often exhibit significant degradation in recognition accuracy when evaluated on another one" [Gonq et al.,ICML 2013, Torralba et al., CVPR 2011, Perronnin et al., CVPR 2010]

- "Hot topic" -Visual domain adaptation [Tutorial CVPR'12, ICCV'13]
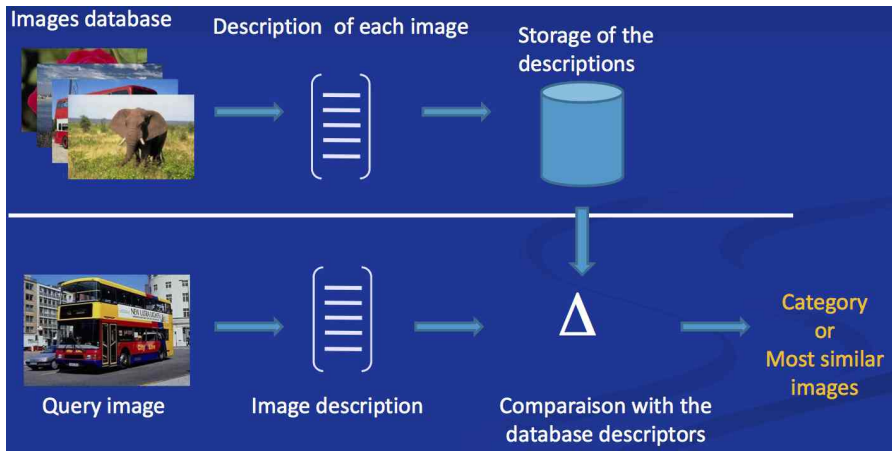
Brief recap on computer vision issues (2) [Slides from J. Sivic]

kmeans
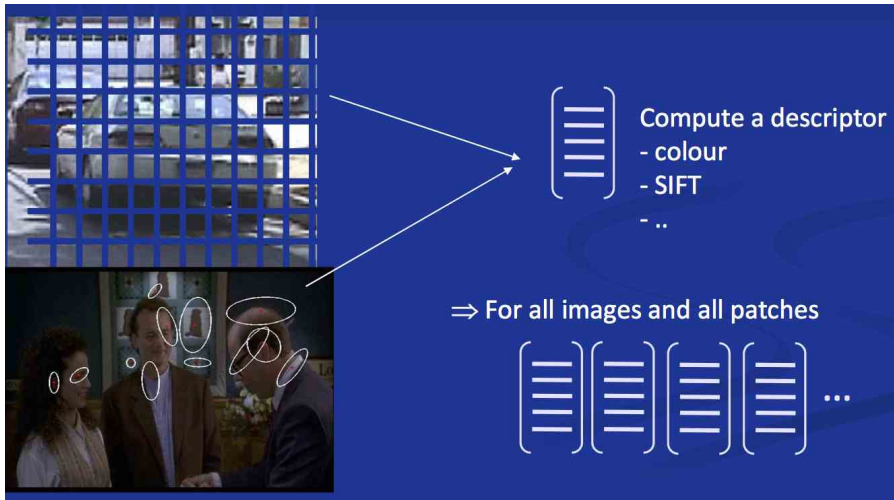
Brief recap on computer vision issues (4) [Slides from J. Sivic]
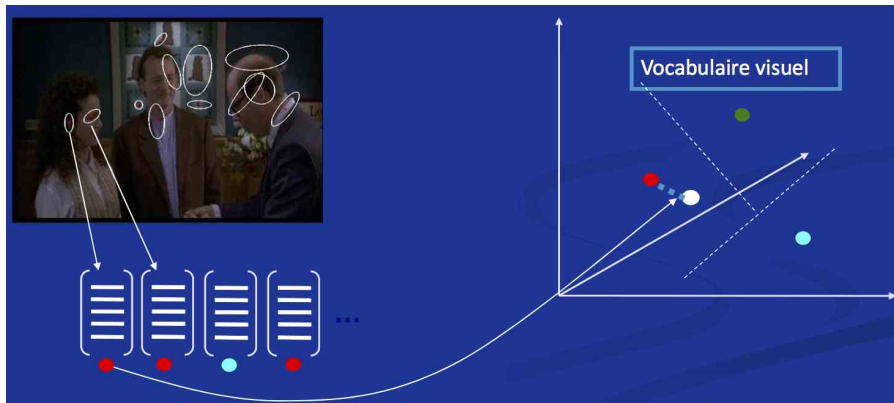
Brief recap on computer vision issues (5) [Slides from J. Sivic]

Word frequency
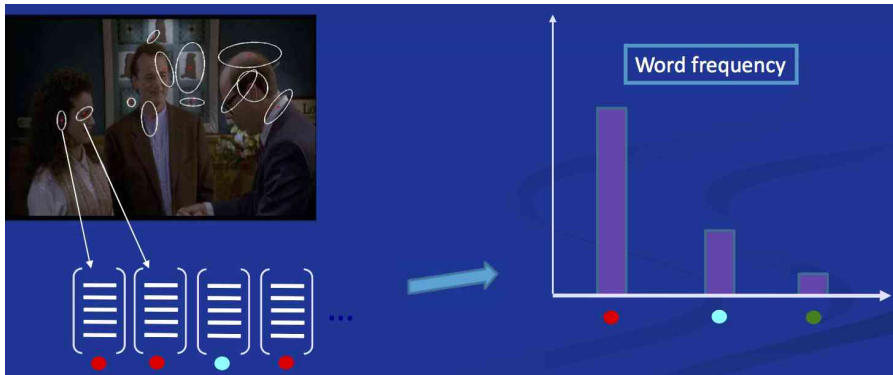
Problems with data representations

digital SLR

webcam

SURF

SIFT

VQ to 300

Different dimensions

VQ to 1000

high quality | low quality | daylight | sunset

posed | "in the wild" | art | surveillance
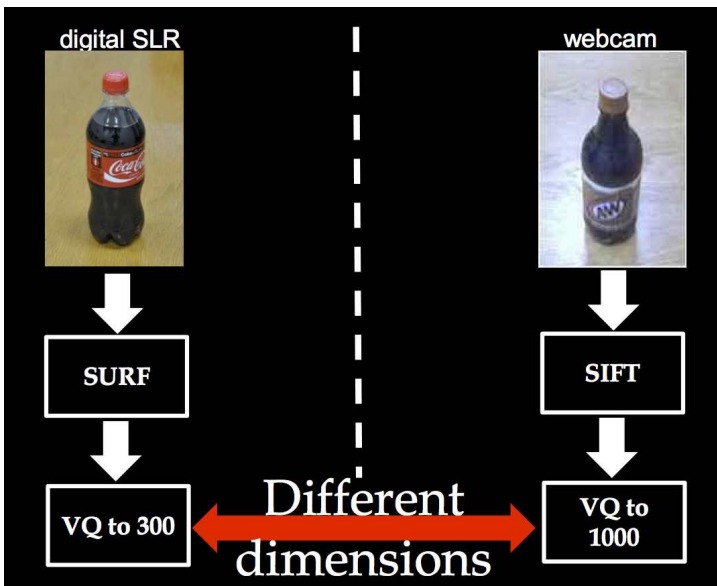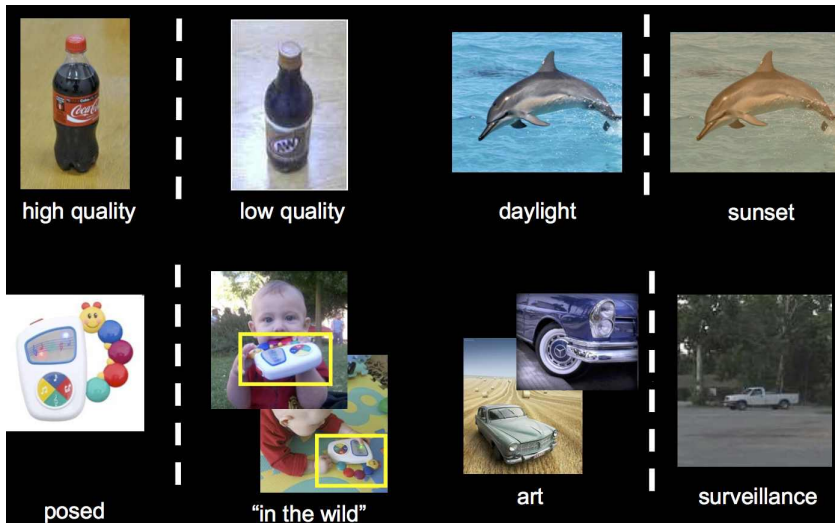
[Xu,Saenko,Tsang, Domain Transfer Tutorial - CVPR'12]

Natural Language Processing

Text are represented by "words" (Bag of Words)

- Part of Speech Tagging: Adapt a tagger learned from medical papers to a journal (Wall Street Journal) - Newsgroup

Biomedical	WSJ
the **signal** *required* to	**investment** *required*
stimulatory **signal** *from*	**buyouts** *from* buyers
essential **signal** *for*	to **jail** *for* violating

- Spam detection: Adapt a classifier from one mailbox to another

Books *Dvds* *Electronics* *Kitchen*

- Sentiment analysis:

Exemple
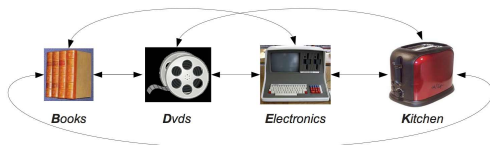
critiques de livres

critiques de film

Domain Adaptation for sentiment analysis - ex [Pan-IJCAI'13 tutorial]

	Electronics	**Video games**
✅	(1) Compact; easy to operate; very good picture quality; looks sharp!	(2) A very good game! It is action packed and full of excitement. I am very much hooked on this game.
✅	(3) I purchased this unit from Circuit City and I was very excited about the quality of the picture. It is really nice and sharp.	(4) Very realistic shooting action and good plots. We played this and were hooked.
❌	(5) It is also quite blurry in very dark settings. I will never_buy HP again.	(6) It is so boring. I am extremely unhappy and will probably never_buy UbiSoft again.

- Source specific: *compact, sharp, blurry*.
- Target specific: *hooked, realistic, boring*.
- Domain independent: *good, excited, nice*, *never_buy, unhappy*.

Other applications

- Speech recognition [Tutorial at Interspeech'12]
- Medecine
- Biology
- Time series
- Wifi localization

Notations

Notations

- $X \subseteq \mathbb{R}^d$ input space, $Y = \{-1, +1\}$ output space
- P_S **source** domain: distribution over $X \times Y$
  D_S marginal distribution over X
- P_T **target** domain: different distribution over $X \times Y$
  D_T marginal distribution over X
- $\mathcal{H} \subseteq Y^X$: hypothesis class

Expected error of a hypothesis $h : X \rightarrow Y$

- $R_{P_S}(h) = \underset{(\mathbf{x}^s, y^s) \sim P_S}{\mathbf{E}} \mathbf{I}\big[h(\mathbf{x}^s) \neq y^s\big]$ **source** domain error
- $R_{P_T}(h) = \underset{(\mathbf{x}^t, y^t) \sim P_T}{\mathbf{E}} \mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big]$ **target** domain error

Domain Adaptation: find $h \in \mathcal{H}$ with R_{P_T} **small** from data $\sim D_T$ and P_S

Classical result in supervised learning

Empirical error

- $R_S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{m_s} \sim (P_S)^{m_s}$ a labeled sample drawn i.i.d. from P_S
- Associated **empirical error** of an hypothesis h:
$$R_S(h) = \frac{1}{m_s} \sum_{i=1}^{m_s} \mathbf{I}\big[h(\mathbf{x}_i^s) \neq y_i^s\big]$$

Classical PAC result: From the same distribution

$$R_{P_S}(h) \leq R_S(h) + O(\frac{complexity(h)}{\sqrt{m_s}})$$

$\Rightarrow$ Occam razor principle

What about R_{P_T} if we have no or very few labeled data? $\rightarrow$ try to make use of source information

Domain Adaptation
Setting

- **Labeled** Source Sample

  $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_s}$ **Source** sample drawn i.i.d. from P_S

- **Unlabeled** Target Sample

  $T = \{\mathbf{x}_j\}_{j=1}^{m_t}$ **Target** Sample drawn i.i.d. from D_T
  optionnal: a few labeled target examples

If h is learned from **source** domain, how does it perform on **target** domain?
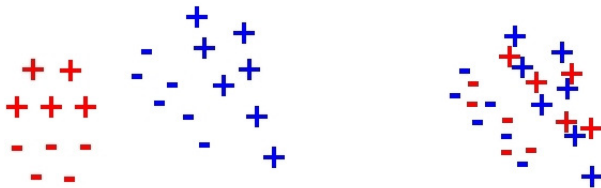
Domain Adaptation
Setting

- **Labeled** Source Sample

  $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m_s}$ **Source** sample drawn i.i.d. from P_S

- **Unlabeled** Target Sample

  $T = \{\mathbf{x}_j\}_{j=1}^{m_t}$ **Target** Sample drawn i.i.d. from D_T
  optionnal: a few labeled target examples

If h is learned from **source** domain, how does it perform on **target** domain?
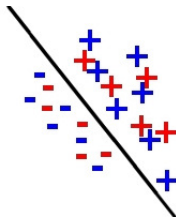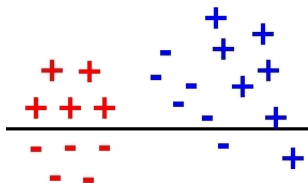
Illustration settings

Classical supervised learning

Apprentissage supervisé

Distribution P_S

Échantillon étiqueté i.i.d. selon P_S → Apprentissage → Modèle

Domain adaptation

Adaptation de domaine

Distribution différente P_T

Distribution P_S

Échantillon non étiqueté i.i.d. selon D_T

Échantillon étiqueté i.i.d. selon P_S → Apprentissage → Modèle
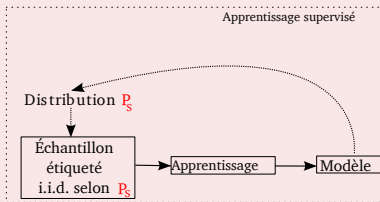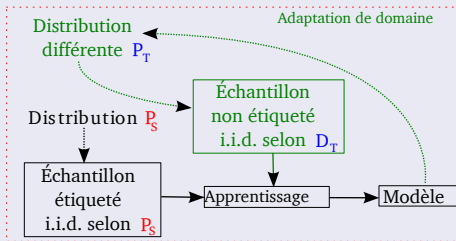
A bit of vocabulary

Unsupervised Transfer Learning
No labels

Unsupervised DA
Presence of source labels, no target labels

Semi-supervised DA
Presence of source labels, few target labels and a lot of unlabeled data

$\neq$ Semi-supervised learning
No distribution shift, few labeled data and a lot of unlabeled data from the same domain

Some key points

- Estimating of the distribution shift
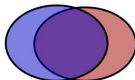
- Deriving generalization guarantees
  $R_{P_T}(h) \leq ? R_{P_S}(h) ? + ?$

Some key points

- Estimating of the distribution shift
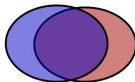
- Deriving generalization guarantees
  $R_{P_T}(h) \leq ? R_{P_S}(h)? + ?$
- Characterizing when the adaptation is possible

Some key points

- Estimating of the distribution shift
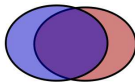
- Deriving generalization guarantees
  $R_{P_T}(h) \leq ?R_{P_S}(h)? + ?$

- Characterizing when the adaptation is possible

- Defining algorithms
  Underlying idea: Try to move closer the two distributions

Some key points

- Estimating of the distribution shift
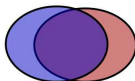
- Deriving generalization guarantees
  $R_{P_T}(h) \leq ? R_{P_S}(h)? + ?$
- Characterizing when the adaptation is possible

- Defining algorithms
  Underlying idea: Try to move closer the two distributions
- Applying model selection principle
  How to tune hyperparameters with no labeled information from target

3 main classes of algorithms

Reweighting/Instance-based methods

Correct a sample bias by reweighting source labeled data:
source instances close to target instances are more important.
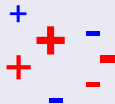
Feature-based methods/Find new representation spaces

Find a common space where source and target are close
(projection, new features, etc)
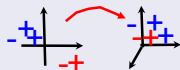
Ajustement/Iterative methods

Modify the model by incorporating pseudo-labeled information

Reweighting/Instance based Methods

Motivation

- Domains share the **same** support (*i.e.* bag of words)
- Distribution shift is caused by **sampling bias**/**shift between marginals**

Idea

Reweight or **select** instances to reduce the discrepancy between source and target domains.
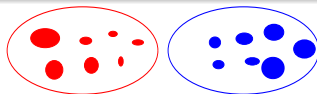
$$R_{P_T}(h) = \mathop{\mathbf{E}}_{(\mathbf{x}^t, y^t) \sim P_T} \mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big]$$

A first analysis

$$R_{P_T}(h) = \mathop{\mathbf{E}}_{(\mathbf{x}^t, y^t) \sim P_T} \mathbf{I}\left[h(\mathbf{x}^t) \neq y^t\right]$$

$$= \mathop{\mathbf{E}}_{(\mathbf{x}^t, y^t) \sim P_T} \frac{P_S(\mathbf{x}^t, y^t)}{P_S(\mathbf{x}^t, y^t)} \mathbf{I}\left[h(\mathbf{x}^t) \neq y^t\right]$$

A first analysis

$$R_{P_T}(h) = \mathop{\mathbf{E}}_{(\mathbf{x}^t, y^t) \sim P_T} \mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big]$$

$$= \mathop{\mathbf{E}}_{(\mathbf{x}^t, y^t) \sim P_T} \frac{P_S(\mathbf{x}^t, y^t)}{P_S(\mathbf{x}^t, y^t)} \mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big]$$

$$= \sum_{(\mathbf{x}^t, y^t)} P_T(\mathbf{x}^t, y^t) \frac{P_S(\mathbf{x}^t, y^t)}{P_S(\mathbf{x}^t, y^t)} \mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big]$$

A first analysis

$$
\begin{aligned}
R_{P_T}(h) &= \mathop{\mathbf{E}}_{(\mathbf{x}^t, y^t) \sim P_T} \mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big] \\
&= \mathop{\mathbf{E}}_{(\mathbf{x}^t, y^t) \sim P_T} \frac{P_S(\mathbf{x}^t, y^t)}{P_S(\mathbf{x}^t, y^t)} \mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big] \\
&= \sum_{(\mathbf{x}^t, y^t)} P_T(\mathbf{x}^t, y^t) \frac{P_S(\mathbf{x}^t, y^t)}{P_S(\mathbf{x}^t, y^t)} \mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big] \\
&= \mathop{\mathbf{E}}_{(\mathbf{x}^t, y^t) \sim P_S} \frac{P_T(\mathbf{x}^t, y^t)}{P_S(\mathbf{x}^t, y^t)} \mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big]
\end{aligned}
$$

Covariate shift [Shimodaira,'00]

$\Rightarrow$ Assume similar tasks, $P_S(y|\mathbf{x}) = P_T(y|\mathbf{x})$, then:

$$\begin{aligned}
&= \underset{(\mathbf{x}^t,y^t)\sim P_S}{\mathbf{E}} \frac{D_T(\mathbf{x}^t)P_T(y^t|\mathbf{x}^t)}{D_S(\mathbf{x}^t)P_S(y^t|\mathbf{x}^t)}\mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big] \\
&= \underset{(\mathbf{x}^t,y^t)\sim P_S}{\mathbf{E}} \frac{D_T(\mathbf{x}^t)}{D_S(\mathbf{x}^t)}\mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big] \\
&= \underset{(\mathbf{x}^t)\sim D_S}{\mathbf{E}} \frac{D_T(\mathbf{x}^t)}{D_S(\mathbf{x}^t)} \underset{y^t\sim P_S(y^t|\mathbf{x}^t)}{\mathbf{E}} \}\mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big]
\end{aligned}$$

$\Rightarrow$ **weighted error** on the source domain: $\omega(x^t) = \frac{D_T(\mathbf{x}^t)}{D_S(\mathbf{x}^t)}$

Idea reweight labeled source data according to an estimate of $\omega(x^t)$:
$$\underset{(\mathbf{x}^t,y^t)\sim P_S}{\mathbf{E}} \omega(\mathbf{x}^t)\mathbf{I}\big[h(\mathbf{x}^t) \neq y^t\big]$$

Illustration

No Bias

$D_S(\mathbf{x}) = D_T(\mathbf{x}) \Rightarrow \omega(\mathbf{x}) = 1$
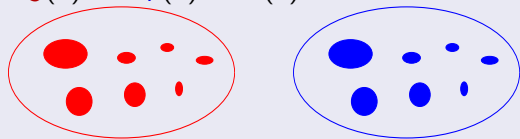
With Bias

$D_S(\mathbf{x}) \neq D_T(\mathbf{x}) \Rightarrow \omega(\mathbf{x}) \neq 1$
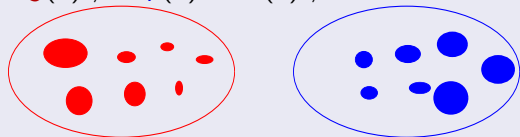
Difficult case

No shared support

$\exists \mathbf{x}, D_S(\mathbf{x}) = 0$ and $D_T(\mathbf{x}) \neq 0$
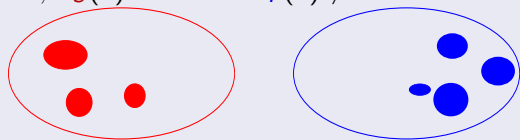
Shared support

$D_S(\mathbf{x}) = 0$ if and only if $D_T(\mathbf{x}) = 0$

Intuition: the quality of the adaptation depends on the magnitude on the weights

How to deal with the sample selection bias?

Setting

A source sample $S = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{m_s}$ and a target sample $T = \{\mathbf{x}_j^t\}_{j=1}^{m_t}$

Estimate new weights without using labels

$$\hat{\omega}(\mathbf{x}_i^s) = \frac{\hat{P}r_T(\mathbf{x}_i^s)}{\hat{P}r_S(\mathbf{x}_i^s)}$$

Learn a classifier on the classifier w.r.t. $\hat{\omega}$

$$\sum_{(\mathbf{x}_i^s, y_i^s) \in S} \hat{\omega}(\mathbf{x}_i^s) I[h(\mathbf{x}_i^s) \neq y_i^s]$$

(Other losses: margin-based hinge-loss, least-square)
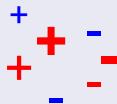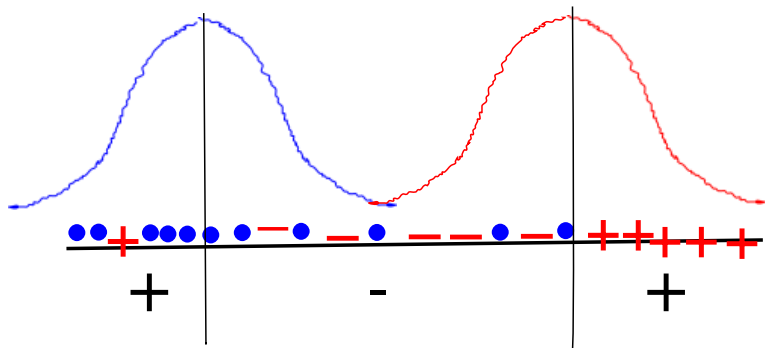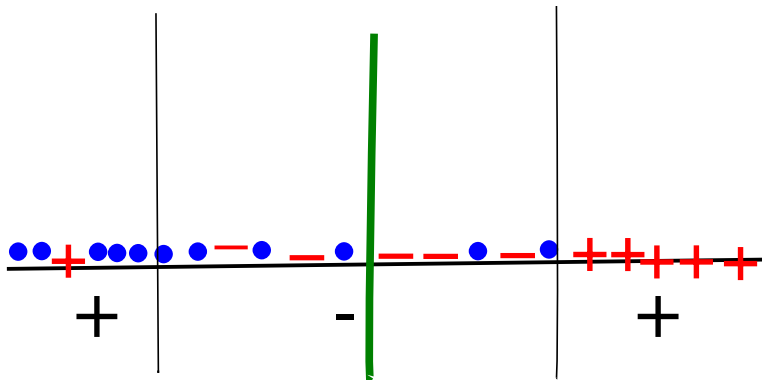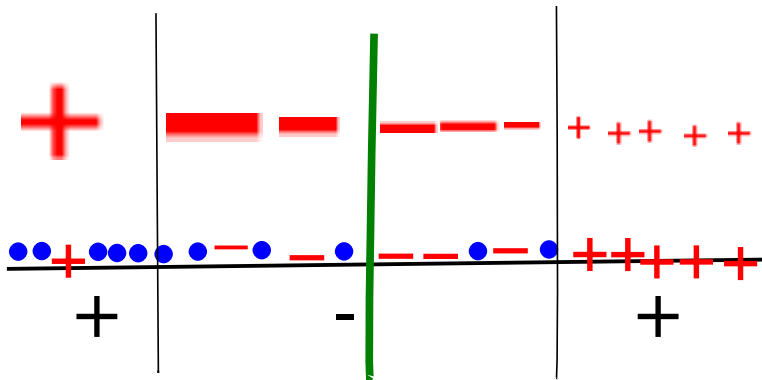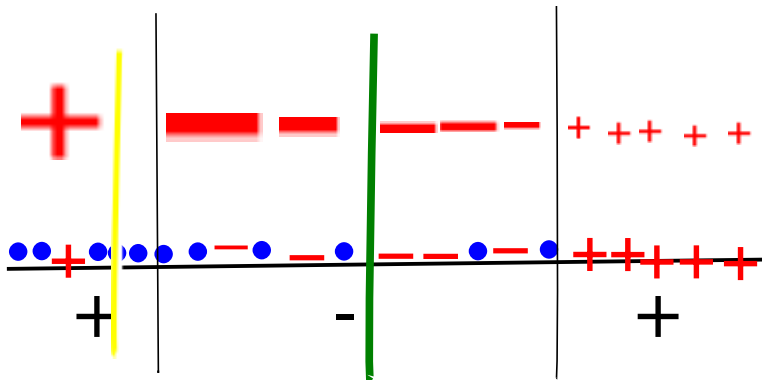
Illustration
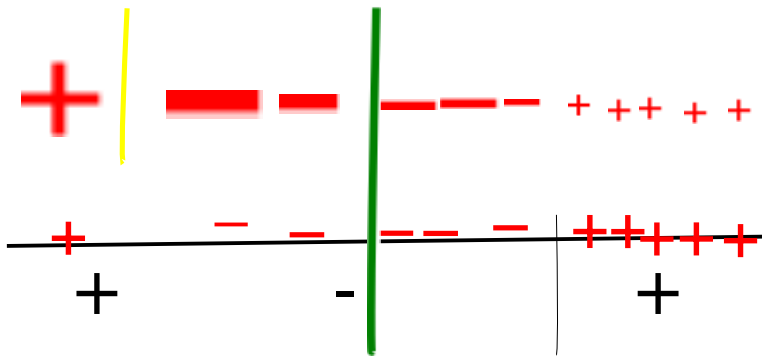
Some existing approaches (1/2)

Density estimators

Build density estimators for source and target domains and estimate the ratio between them - Ex [Sugiyama et al.,NIPS'07]:

- $\hat{\omega}(\mathbf{x}) = \sum_{l=1}^{b} \alpha_l \psi_l(\mathbf{x})$
- Learning: $\operatorname{argmin}_\alpha KL(\hat{\omega} D_S, D_T)$

Learn the weights discriminatively [Bickel et al.,ICML'07]

- Assume $\frac{D_T(\mathbf{x}_i)}{D_S(\mathbf{x}_i)} \propto \frac{1}{p(q=1|\mathbf{x},\theta)}$
- Label source with label 1, target with label 0 and train a classifier ($\hat{\theta}$) to classify examples 1 or 0 (e.g. with logistic regression)
- Compute the new weights $\hat{\omega}(\mathbf{x}_i^s) = \dfrac{1}{p(q=1|\mathbf{x}_i^s; \hat{\theta})}$

Some existing approaches (2/2)

Kernel Mean Matching [Huang et al.,NIPS'06]

- Maximum Mean Discrepancy
  $\text{MMD}(S, T) = \|\frac{1}{m_S} \sum_{i=1}^{m_s} \phi(\mathbf{x}_i^s) - \frac{1}{m_t} \sum_{j=1}^{m_t} \phi(\mathbf{x}_j^t)\|_H$

- $\min_\beta \|\frac{1}{m_S} \sum_{i=1}^{m_s} \beta(\mathbf{x}_i^s)\phi(\mathbf{x}_i^s) - \frac{1}{m_S} \sum_{j=1}^{m_t} \phi(\mathbf{x}_j^t)\|_H$
  s.t. $\beta(\mathbf{x}_i^s) \in [0, B]$ and $|\frac{1}{m_S} \sum_{i=1}^{m_s} \beta(\mathbf{x}_i^s) - 1| < \epsilon$

- $\min_\beta \frac{1}{2}\beta^T \mathbf{K}_T \beta - \kappa_{S,T}^T$ s.t. $\beta_i \in [0, B]$ and $|\sum_{i=1}^{m_s} \beta(\mathbf{x}_i^s) - m_s| < m_s\epsilon$

Guarantees [Gretton et al.,2008] - Under covariate shift assumptions

$|R_{P_T}(h) - \text{weighted}(R_S(h))| < \sqrt{\frac{O(1/\delta) + O(\max_{\mathbf{x}} \omega(\mathbf{x})^2)}{m_s}} + C\epsilon$ and

$\|\frac{1}{m_S} \sum_{i=1}^{m_s} \omega(\mathbf{x}_i^s)\phi(\mathbf{x}_i^s) - \frac{1}{m_t} \sum_{j=1}^{m_t} \phi(\mathbf{x}_j^t)\| \leq O((1/\delta)\sqrt{\omega_{\max}^2/m_s + 1/m_t})$

Bad news

- DA is hard, even under covariate shift [Ben-David et al.,ALT'12]
  $\Rightarrow$ To learn a classifier the number of examples depend on $|\mathcal{H}|$ (finite) or exponentially on the dimension of X
- Covariate shift assumption may fail: Tasks are not similar in general $P_S(y|\mathbf{x}) \neq P_T(y|\mathbf{x})$
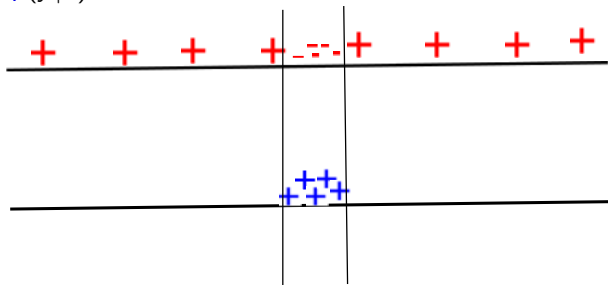
- We did not consider the hypothesis space.
- Can define a general theory about DA?

Theoretical frameworks for Domain Adaptation

A keypoint: estimating the distribution shift

First idea: Total variation measure

$d_{L_1}(D_S, D_T) = \sup_{B \subseteq X} |D_S(B) - D_T(B)|$

Subset of points maximizing the divergence

But:
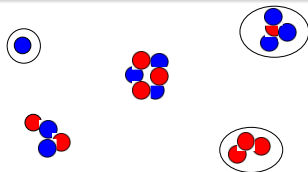
- Not computable in general, and thus not estimable from finite samples

A keypoint: estimating the distribution shift

First idea: Total variation measure

$d_{L_1}(D_S, D_T) = \sup_{B \subseteq X} |D_S(B) - D_T(B)|$

Subset of points maximizing the divergence

But:

- Not computable in general, and thus not estimable from finite samples
- Not related to the hypothesis class
- Do not characterize the difficulty of the problem for $\mathcal{H}$

The $\mathcal{H}\Delta\mathcal{H}$-divergence [Ben-David et al.,NIPS'06;MLJ'10]

Definition

$$d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) = \sup_{(h,h')\in\mathcal{H}^2} \left| R_{D_T}(h, h') - R_{D_S}(h, h') \right|$$

$$= \sup_{(h,h')\in\mathcal{H}^2} \left| \mathop{\mathbf{E}}_{\mathbf{x}^t\sim D_T} \mathbf{I}\big[h(\mathbf{x}^t)\neq h'(\mathbf{x}^t)\big] - \mathop{\mathbf{E}}_{\mathbf{x}^s\sim D_S} \mathbf{I}\big[h(\mathbf{x}^s)\neq h'(\mathbf{x}^s)\big] \right|$$

The $\mathcal{H}\Delta\mathcal{H}$-divergence [Ben-David et al.,NIPS'06;MLJ'10]

Definition

$$d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) = \sup_{(h,h')\in\mathcal{H}^2} \left| R_{D_T}(h,h') - R_{D_S}(h,h') \right|$$

$$= \sup_{(h,h')\in\mathcal{H}^2} \left| \mathop{\mathbf{E}}_{\mathbf{x}^t \sim D_T} \mathbf{I}\big[h(\mathbf{x}^t) \neq h'(\mathbf{x}^t)\big] - \mathop{\mathbf{E}}_{\mathbf{x}^s \sim D_S} \mathbf{I}\big[h(\mathbf{x}^s) \neq h'(\mathbf{x}^s)\big] \right|$$

Illustration with **only 2 hypothesis in** $\mathcal{H}$ h and h'
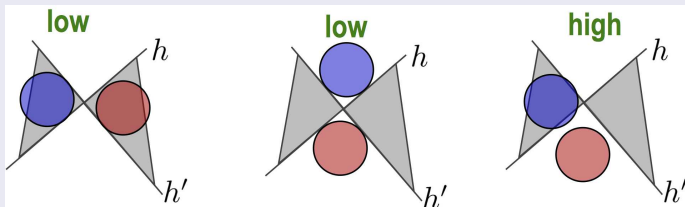
Note: With a larger $\mathcal{H}$, the distance will be **high** since we can easily find two hypothesis able to **distinguish** the two domains

Computable from samples

Consider two samples S, T of size m from D_S and D_T

$$d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) \leq d_{\mathcal{H}\Delta\mathcal{H}}(S, T) + O(\text{complexity}(\mathcal{H})\sqrt{\tfrac{\log(m)}{m}})$$

complexity($\mathcal{H}$): VC-dimension [Ben-david et al.,06;'10], Rademacher [Mansour et al.,'09]

Empirical estimation

$$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(S, T) = 2\left(1 - min_{h\in\mathcal{H}}\left[\frac{1}{m}\sum_{\mathbf{x}:h(\mathbf{x})=-1} I[\mathbf{x} \in S] + \frac{1}{m}\sum_{\mathbf{x}:h(\mathbf{x})=1} I[\mathbf{x} \in T]\right]\right)$$

$\Rightarrow$ Already seen: label source examples as -1, target ones as +1 and try to learn a classifier in $\mathcal{H}$ minimizing the associated empirical error

Going to a generalization bound

Preliminaries

- $R_{P_T}(h, h') = \underset{(\mathbf{x}, y) \sim P_S}{\mathbf{E}} I[h(\mathbf{x}) \neq h'(\mathbf{x})] = \underset{\mathbf{x} \sim D_T}{\mathbf{E}} I[h(\mathbf{x}) \neq h'(\mathbf{x})]$

  R_{P_T} (R_{P_S}) fulfills the triangle inequality

- $|R_{P_T}(h, h') - R_{P_S}(h, h')| \leq \frac{1}{2} d_{\mathcal{H} \Delta \mathcal{H}}(D_S, D_T)$

  since $d_{\mathcal{H} \Delta \mathcal{H}}(D_S, D_T) = 2 \sup_{(h, h') \in \mathcal{H}^2} \left| R_{D_T}(h, h') - R_{D_S}(h, h') \right|$

- $h_S^* = \operatorname{argmin}_{h \in \mathcal{H}} R_{P_S}(h)$: best on source

- $h_T^* = \operatorname{argmin}_{h \in \mathcal{H}} R_{P_T}(h)$: best on target

Ideal joint hypothesis

$h^* = \operatorname{argmin}_{h \in \mathcal{H}} R_{P_S}(h) + R_{P_T}(h)$; $\lambda = R_{P_S}(h^*) + R_{P_T}(h^*)$

A first bound

$$R_{P_T}(h) \leq$$

A first bound

$$R_{P_T}(h) \leq R_{P_T}(h^*) + R_{P_T}(h, h^*)$$

A first bound

$$R_{P_T}(h) \leq R_{P_T}(h^*) + R_{P_T}(h, h^*)$$
$$\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + R_{P_T}(h, h^*) - R_{P_S}(h, h^*)$$

A first bound

$$R_{P_T}(h) \leq R_{P_T}(h^*) + R_{P_T}(h, h^*)$$
$$\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + R_{P_T}(h, h^*) - R_{P_S}(h, h^*)$$
$$\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + |R_{P_T}(h, h^*) - R_{P_S}(h, h^*)|$$

A first bound

$$R_{P_T}(h) \leq R_{P_T}(h^*) + R_{P_T}(h, h^*)$$

$$\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + R_{P_T}(h, h^*) - R_{P_S}(h, h^*)$$

$$\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + |R_{P_T}(h, h^*) - R_{P_S}(h, h^*)|$$

$$\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + \frac{1}{2} d_{\mathcal{H} \triangle \mathcal{H}}(D_S, D_T)$$

A first bound

$$\begin{aligned}
R_{P_T}(h) &\leq R_{P_T}(h^*) + R_{P_T}(h, h^*) \\
&\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + R_{P_T}(h, h^*) - R_{P_S}(h, h^*) \\
&\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + |R_{P_T}(h, h^*) - R_{P_S}(h, h^*)| \\
&\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) \\
&\leq R_{P_T}(h^*) + R_{P_S}(h) + R_{P_S}(h^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)
\end{aligned}$$

A first bound

$$
\begin{aligned}
R_{P_T}(h) &\leq R_{P_T}(h^*) + R_{P_T}(h, h^*) \\
&\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + R_{P_T}(h, h^*) - R_{P_S}(h, h^*) \\
&\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + |R_{P_T}(h, h^*) - R_{P_S}(h, h^*)| \\
&\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) \\
&\leq R_{P_T}(h^*) + R_{P_S}(h) + R_{P_S}(h^*) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) \\
&\leq R_{P_S}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \lambda
\end{aligned}
$$

$$R_{P_T}(h) \leq R_{P_T}(h^*) + R_{P_T}(h, h^*)$$

$$\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + R_{P_T}(h, h^*) - R_{P_S}(h, h^*)$$

$$\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + |R_{P_T}(h, h^*) - R_{P_S}(h, h^*)|$$

$$\leq R_{P_T}(h^*) + R_{P_S}(h, h^*) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$$

$$\leq R_{P_T}(h^*) + R_{P_S}(h) + R_{P_S}(h^*) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$$

$$\leq R_{P_S}(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \lambda$$

$$\leq R_S(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(S, T) + O(\text{complexity}(\mathcal{H})\sqrt{\frac{\log(m)}{m}}) + \lambda$$

Main theoretical bound

Theorem [Ben-David et al.,MLJ'10,NIPS'06]

Let $\mathcal{H}$ a symmetric hypothesis space. If D_S and D_T are respectively the marginal distributions of source and target instances, then for all $\delta \in (0, 1]$, with probability at least $1 - \delta$:

$$\forall h \in \mathcal{H}, \quad R_{P_T}(h) \leq R_{P_S}(h) + \frac{1}{2} d_{\mathcal{H} \Delta \mathcal{H}}(D_S, D_T) + \lambda$$

Formalizes a natural approach: Move closer the two distributions while ensuring a low error on the source domain.

Justifies many algorithms:

- reweighting methods,
- feature-based methods,
- adjusting/iterative methods.

$$R_{P_T}(h) \leq$$

$$R_{P_T}(h) \leq R_{P_T}(h, h_S^*) + R_{P_T}(h_S^*, h_T^*) + R_{P_T}(h_T^*)$$

$$R_{P_T}(h) \leq R_{P_T}(h, h_S^*) + R_{P_T}(h_S^*, h_T^*) + R_{P_T}(h_T^*)$$
$$= R_{P_T}(h, h_S^*) + \nu$$

$$R_{P_T}(h) \leq R_{P_T}(h, h_S^*) + R_{P_T}(h_S^*, h_T^*) + R_{P_T}(h_T^*)$$
$$= R_{P_T}(h, h_S^*) + \nu$$
$$\leq R_{P_S}(h, h_S^*) + R_{P_T}(h, h_S^*) - R_{P_S}(h, h_S^*) + \nu$$

$$R_{P_T}(h) \leq R_{P_T}(h, h_S^*) + R_{P_T}(h_S^*, h_T^*) + R_{P_T}(h_T^*)$$
$$= R_{P_T}(h, h_S^*) + \nu$$
$$\leq R_{P_S}(h, h_S^*) + R_{P_T}(h, h_S^*) - R_{P_S}(h, h_S^*) + \nu$$
$$\leq R_{P_S}(h, h_S^*) + |R_{P_T}(h, h_S^*) - R_{P_S}(h, h_S^*)| + \nu$$

Another analysis [Mansour et al.,COLT'09]

$$R_{P_T}(h) \leq R_{P_T}(h, h_S^*) + R_{P_T}(h_S^*, h_T^*) + R_{P_T}(h_T^*)$$
$$= R_{P_T}(h, h_S^*) + \nu$$
$$\leq R_{P_S}(h, h_S^*) + R_{P_T}(h, h_S^*) - R_{P_S}(h, h_S^*) + \nu$$
$$\leq R_{P_S}(h, h_S^*) + |R_{P_T}(h, h_S^*) - R_{P_S}(h, h_S^*)| + \nu$$
$$\leq R_{P_S}(h, h_S^*)) + \frac{1}{2} d_{\mathcal{H} \Delta \mathcal{H}}(D_S, D_T) + \nu$$

$$R_{P_T}(h) \leq R_{P_T}(h, h_S^*) + R_{P_T}(h_S^*, h_T^*) + R_{P_T}(h_T^*)$$
$$= R_{P_T}(h, h_S^*) + \nu$$
$$\leq R_{P_S}(h, h_S^*) + R_{P_T}(h, h_S^*) - R_{P_S}(h, h_S^*) + \nu$$
$$\leq R_{P_S}(h, h_S^*) + |R_{P_T}(h, h_S^*) - R_{P_S}(h, h_S^*)| + \nu$$
$$\leq R_{P_S}(h, h_S^*)) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \nu$$
$$(\leq R_{P_S}(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + R_{P_S}(h_S^*) + \nu) \text{ if } h_S^* \text{ is not the true lab}$$

Another analysis [Mansour et al.,COLT'09]

$$R_{P_T}(h) \leq R_{P_T}(h, h_S^*) + R_{P_T}(h_S^*, h_T^*) + R_{P_T}(h_T^*)$$
$$= R_{P_T}(h, h_S^*) + \nu$$
$$\leq R_{P_S}(h, h_S^*) + R_{P_T}(h, h_S^*) - R_{P_S}(h, h_S^*) + \nu$$
$$\leq R_{P_S}(h, h_S^*) + |R_{P_T}(h, h_S^*) - R_{P_S}(h, h_S^*)| + \nu$$
$$\leq R_{P_S}(h, h_S^*)) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + \nu$$
$$(\leq R_{P_S}(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T) + R_{P_S}(h_S^*) + \nu) \text{ if } h_S^* \text{ is not the true lab}$$

- This analysis can lead to smaller when adaptation is possible
- Leads to the same type of bound, just the constant changes .....

Characterization of the possibility of domain adaptation

Constants characterize when adaptation is possible

- $\lambda = R_{P_S}(h^*) + R_{P_T}(h^*)$, $h^* = \mathrm{argmin}_{h \in \mathcal{H}} R_{P_S}(h) + RPT(h)$
  There must exist an ideal joint hypothesis with small error

- $\nu = R_{P_T}(h_S^*, h_T^*) + R_{P_T}(h_T^*)$
  there must exist a very good hypothesis on the target and the best hypothesis on source must be close to the best on target w.r.t to D_T

Other settings

Discrepancy [Mansour et al.,COLT'09]

- instead of the 0-1 loss, more general loss functions ℓ (*i.e.* L_p norms)
  $\text{disc}^\ell(D_S, D_T) = \sup_{h,h' \in \mathcal{H}} |R_{D_S}^\ell(h, h') - R_{D_T}^\ell(h, h')|$
- This discrepancy can be minimized and used as a reweighting method
  ([Mansour et al.,COLT'09] - polynomial for L_2 norm for example)

Using some target labeled data

- Weighting the empirical source and target risks [Ben David et al.,2010]
- Using a divergence taking into account target labels [Zhang et al.,NIPS'12] (a divergence must take into account marginals over X and Y, the λ constant counts for Y)

Other settings

Averaged quantities [Germain et al.,ICML'13]

- Consider a probability distribution ρ (posterior) over $\mathcal{H}$ to learn and the following risk: $\underset{h\sim\rho}{\mathbf{E}}\ R_{P_T}(h)$

- Definition of an averaged distance
$$\text{dis}(D_S, D_T) = \left| \underset{h,h'\sim\rho^2}{\mathbf{E}}[R_{D_T}(h, h') - R_{D_S}(h, h')] \right|$$

- Similar generalization bound
$$\underset{h\sim\rho}{\mathbf{E}}\ R_{P_T}(h) \leq \underset{h\sim\rho}{\mathbf{E}}\ R_{P_S}(h) + \text{dis}(D_S, D_T) + \lambda_{\rho^*}$$

- Estimation from samples controlled by PAC-Bayesian theory

- Bound tighter without a supremum

Feature/Projection based Approaches

- Change the feature representation X to better represent shared characteristics between the two domains
  - some features are domain-specific,
  - others are generalizable
  - or there exist mappings from the original space
- $\Rightarrow$ Make source and target domain explicitly similar
- $\Rightarrow$ Learn a new feature space by embedding or projection

Source　　　　Target

joint feature space
(smaller or higher dimension)
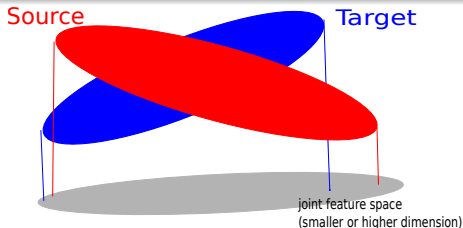
Metric Learning [Kulis et al.,'11;Saenko et al.,'10]

- Mahalanobis: $d_{\mathbf{W}}^2(\mathbf{x}, \mathbf{x}') = (\mathbf{x} - \mathbf{x}')^T \mathbf{W}(\mathbf{x} - \mathbf{x}')$

- PSD matrix $\mathbf{W} = L^T L$,
  L projection space of dimension $\mathbb{R}^{rank(\mathbf{W}) \times d}$
  $(L\mathbf{x} - L\mathbf{x}')^T (L\mathbf{x} - L\mathbf{x}')$

- Pair-wise constraints: source ex. $(\mathbf{x}_i^s, y_i^s)$ and target $(\mathbf{x}_j^t, y_j^t)$
  - $d_{\mathbf{W}}^2(\mathbf{x}_i^s, \mathbf{x}_j^t) \leq u$ if $y_i^s = y_j^t$ (source and target must be similar)
  - $d_{\mathbf{W}}^2(\mathbf{x}_i^s, \mathbf{x}_j^t) \geq l$ if $y_i^s \neq y_j^t$ (source and target must be dissimilar)
  - Require some target labels

Metric Learning [Kulis et al.,CVPR'11;Saenko et al.,ECCV'10]

(a) Domain shift problem   (b) Pairwise constraints   (c) Invariant space

[Saenko et al.,ECCV'10]

Formulation (based on ITML [Davis et al.,ICML'07])

$$\min_{\mathbf{W} \succeq 0} \quad Tr(\mathbf{W}) - \log \det \mathbf{W}$$

$$\text{s.t.} \quad d_{\mathbf{W}}^2(\mathbf{x}_i^s, \mathbf{x}_j^t) \leq u, \forall (\mathbf{x}_i^s, \mathbf{x}_j^t) \in \mathcal{S}imilarSet$$

$$d_{\mathbf{W}}^2(\mathbf{x}_i^s, \mathbf{x}_j^t) \geq l, \forall (\mathbf{x}_i^s, \mathbf{x}_j^t) \in \mathcal{D}issimilarSet$$

$\Rightarrow$ Can be kernelized

(Simple) Feature augmentation [Daume III et al.,'07;'10]

- $\phi(\mathbf{x}) = <\mathbf{x}, \mathbf{x}, 0>$ for source instances
- $\phi(\mathbf{x}) = <\mathbf{x}, 0, \mathbf{x}>$ for target instances
- $\Rightarrow$ Share some relevant features and not irrelevant ones (*e.g.* in text sentiment analysis: find shared words)
  $\Rightarrow$ a way to allow the existency of the ideal joint hypothesis h^*

Learn in the new space ϕ

- Require target labels
- Bound: $R_{D_T} \leq \frac{1}{2}(R_T + R_S) + O(complexity) + (\frac{1}{m_s} + \frac{1}{m_t})O(\frac{1}{\delta}) + O(d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T))$
- Kernelized and semi-supervised versions [add: $(+1, <0, \mathbf{x}, -\mathbf{x}>)$ and $(-1, <0, \mathbf{x}, -\mathbf{x}>)$ to learning sample]

Find latent spaces - Structural Correspondence Learning [Blitzer et al.,'07]

Identify shared features

Domains	Negative	Positive
Books	plot <num>_pages predictable reading_this page_<num>	reader grisham engaging must_read fascinating
Kitchen	the_plastic poorly_designed leaking awkward_to defective	excellent_product espresso are_perfect years_now a_breeze
Pivot features	weak don't_waste awful	and_easy loved_it a_wonderful a_must highly_recommended

- Sentiment analysis - Bag of words (bigrams)
- Choose K pivot features (frequent words in both domains, highly correlated with labels)
- Learn K classifiers to predict pivot features from remaining features
- For each feature add K new features
- Represents source and target data with these features

Find latent spaces - Structural Correspondence Learning [Blitzer et al.,'07]

- Apply PCA source+target new features to get a low rank latent representation
- Learn a classifier in the new projection space defined by PCA

Source

Target

$P\boldsymbol{x}$

$P\boldsymbol{x}$
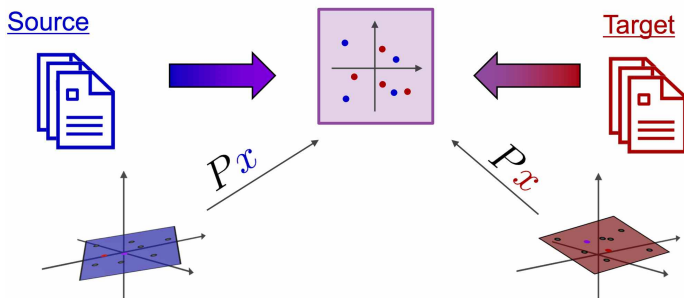
Manifold-based methods
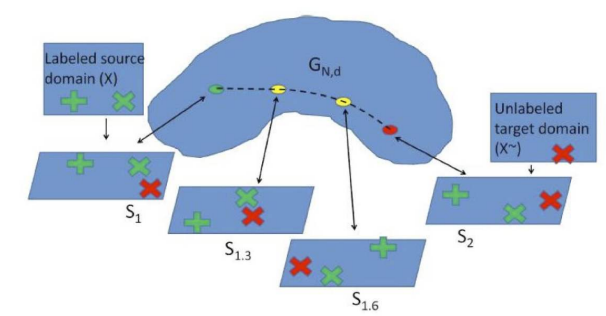
- Assume $X \subseteq \mathbb{R}^N$
- Apply PCA on source data $\Rightarrow$ matrix $\mathbf{S}_1$ of rank d
- Apply PCA on target data $\Rightarrow$ matrix $\mathbf{S}_2$ of rank d
- Geodesic path on the Grassman manifold $\mathbb{G}_{N,d}$ (d-dimensional vector subspaces $\subset \mathbb{R}^N$) between $\mathbf{S}_1$ and $\mathbf{S}_2$

Manifold-based methods
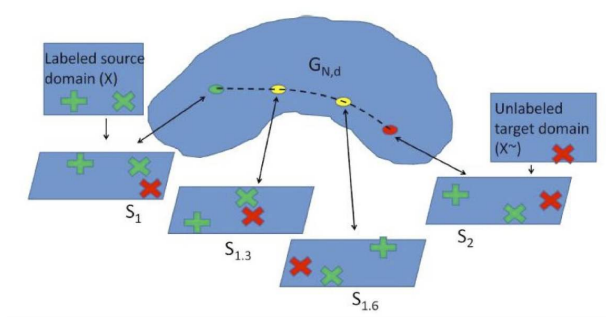
[Gopalan et al.,'10]

- Use of an exponential flow $\psi(t') = \mathbf{Q}\exp(t'\mathbf{B})\mathbf{J}$
  with $\mathbf{Q}$ $N \times N$ matrix with determinant 1 s.t. $\mathbf{Q}^T\mathbf{S}_1 = \mathbf{J}$ and $\mathbf{J}^T = [\mathbf{I}_d\mathbf{0}_{N-d,d}]$
  intermediate subspaces are obtained by computing $\mathbf{B}$ (skew block-diagonal matrix)
  and varying t' between 0 and 1
- Take a collection S' of l subspaces between $\mathbf{S}_1$ and $\mathbf{S}_2$ on the manifold
- Project the data on $\mathbf{S}'$ and learn in that new space

A simpler approach - Subspace alignment [Fernando et al.,ICCV'13]
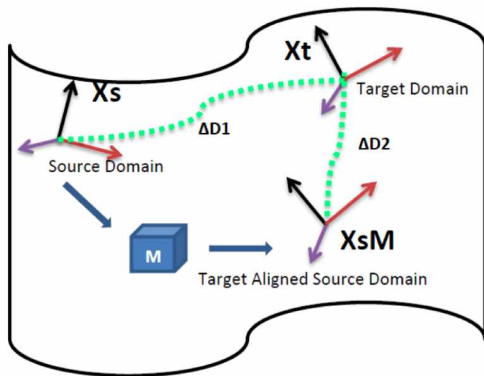
- Move closer PCA-based representations
- Totally unsupervised

Subspace alignment algorithm

Algorithm 1: Subspace alignment DA algorithm

Data: Source data S, Target data T, Source labels Y_S, Subspace dimension d
Result: Predicted target labels Y_T
$\mathbf{S}_1 \leftarrow PCA(S, d)$ (source subspace defined by the first d eigenvectors) ;
$\mathbf{S}_2 \leftarrow PCA(T, d)$ (target subspace defined by the first d eigenvectors);
$\mathbf{X}_a \leftarrow \mathbf{S}_1 \mathbf{S}_1{}' \mathbf{S}_2$ (operator for aligning the source subspace to the target one);
$\mathbf{S}_a = S\mathbf{X}_a$ (new source data in the aligned space);
$\mathbf{T}_T = T\mathbf{S}_2$ (new target data in the aligned space);
$Y_T \leftarrow Classifier(\mathbf{S}_a, \mathbf{T}_T, Y_S)$;

- $\mathbf{M}^* = \mathbf{S}_1{}'\mathbf{S}_2$ corresponds to the "subspace alignment matrix":
  $\mathbf{M}^* = \mathrm{argmin}_\mathbf{M} \|\mathbf{S}_1\mathbf{M} - \mathbf{S}_2\|$
- $X_a = \mathbf{S}_1\mathbf{S}_1{}'\mathbf{S}_2 = \mathbf{S}_1\mathbf{M}^*$ projects the source data to the target subspace
- A natural similarity: $\mathrm{Sim}(\mathbf{x}_s, \mathbf{x}_t) = \mathbf{x}_s\mathbf{S}_1\mathbf{M}^*\mathbf{S}_1{}'\mathbf{x}_t' = \mathbf{x}_s\mathbf{A}\mathbf{x}_t'$

Some results

- Adaptation from Office/Caltech-10 datasets (four domains to adapt) is used as source and one as target
- Comparisons
  - Baseline 1: projection on the source subspace
  - Baseline 2: projection on the target subspace
  - 2 related methods : GFK [Gong et al., CVPR'12] and GFS [Gopalan et al.,ICCV'11]

Some results

- Office/Caltech-10 datasets with 4 domains A, B, C ,D

Method	C→A	D→A	W→A	A→C	D→C	W→C
NA	21.5	26.9	20.8	22.8	24.8	16.4
Baseline 1	38.0	29.8	35.5	30.9	29.6	31.3
Baseline 2	**40.5**	33.0	**38.0**	33.3	31.2	31.9
GFS [8]	36.9	32	27.5	35.3	29.4	21.7
GFK [7]	36.9	32.5	31.1	**35.6**	29.8	27.2
OUR	39.0	**38.0**	37.4	35.3	**32.4**	**32.3**

Method	A→D	C→D	W→D	A→W	C→W	D→W
NA	22.4	21.7	40.5	23.3	20.0	53.0
Baseline 1	34.6	37.4	71.8	35.1	33.5	74.0
Baseline 2	34.7	36.4	72.9	36.8	34.4	78.4
GFS [8]	30.7	32.6	54.3	31.0	30.6	66.0
GFK [7]	35.2	35.2	70.6	34.4	33.7	74.9
OUR	**37.6**	**39.6**	**80.3**	**38.6**	**36.8**	**83.6**

Table 2. Recognition accuracy with unsupervised DA using a NN classifier (Office dataset + Caltech10).

Method	C→A	D→A	W→A	A→C	D→C	W→C
Baseline 1	44.3	36.8	32.9	36.8	29.6	24.9
Baseline 2	44.5	38.6	34.2	37.3	31.6	28.4
GFK	44.8	37.9	37.1	38.3	31.4	29.1
OUR	**46.1**	**42.0**	**39.3**	**39.9**	**35.0**	**31.8**

Method	A→D	C→D	W→D	A→W	C→W	D→W
Baseline 1	36.1	38.9	73.6	**42.5**	34.6	75.4
Baseline 2	32.5	35.3	73.6	37.3	34.2	80.5
GFK	37.9	36.1	74.6	39.8	34.9	79.1
OUR	**38.8**	**39.4**	**77.9**	39.6	**38.9**	**82.3**

Table 3. Recognition accuracy with unsupervised DA using a SVM classifier(Office dataset + Caltech10).

- Divergences

Method	NA	Baseline 1	Baseline 2	GFK	SA
TDAS	1.25	3.34	2.74	2.84	**4.26**
HΔH	98.1	99.0	99.0	74.3	**53.2**

Feature-based method

- Feature-based approaches are very popular
  Many other (SVM/kernel-based, MKL, deep learning [Glorot et al.,ICML'11], ...) methods not covered here,
- Subspace-based methods $\Rightarrow$ "hot topic"
- Embed similarity map: define feature as similarity to landmarks points
  - labeled source instances distributed similarly to the target domain

Source   Target $\Rightarrow$

[Grauman,VisDA-WS_ICCV'13] $\rightarrow$ subsampling: work with instances facilitating adaptation
or use distances to headphones as a representation
$< k(\cdot, \mathbf{x}_1), k(\cdot, \mathbf{x}_2), k(\cdot, \mathbf{x}_3), k(\cdot, \mathbf{x}_4), k(\cdot, \mathbf{x}_5), ... >, ...$

Adjusting/Iterative methods

Principle

- Integrate some information about the target samples iteratively
  $\Rightarrow$ use of pseudo-labels
- "Move" closer distributions
  $\Rightarrow$ Remove/add some instances $\Rightarrow$ take into account a divergence measure
- Repeat the process until convergence or no remaining instances

DASVM [Bruzzone et al.,'10]

A brief recap on SVM

- Learning sample $LS = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$
- Learn a classifier $h(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b$

$$\text{Formulation:} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^n \xi_i$$

$$\text{subject to} \quad \ell_i(\langle \mathbf{w}, \mathbf{x_i} \rangle + b) \geq 1 - \xi_i, \quad 1 \leq i \leq n$$

$$\boldsymbol{\xi} \succeq 0$$
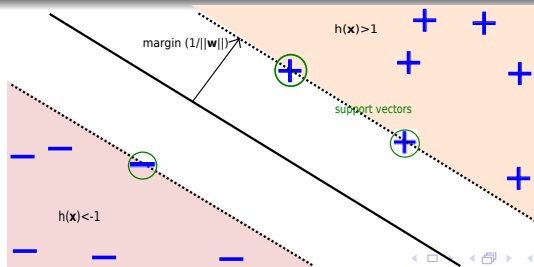
DASVM principle

1. $LS = S$
2. Learn a classifier h^0 from the learning sample LS
3. Repeat until stopping criterion
   - Select the first k target examples $\mathbf{x}^t$ s.t. $0 < h(\mathbf{x}^t) < 1$ with highest margin and affect the pseudo-label -1
   - Select the first k target examples $\mathbf{x}^t$ s.t. $-1 < h(\mathbf{x}^t) < 0$ with highest margin and affect them the pseudo-label $+1$
   - Add these $2k$ examples (pseudo-labeled) to LS
   - Remove from LS the first k positive and k negative source instances with highest margin
4. Output the last classifier

Algorithm stops when the number of selected instances at each step downs to a threshold.
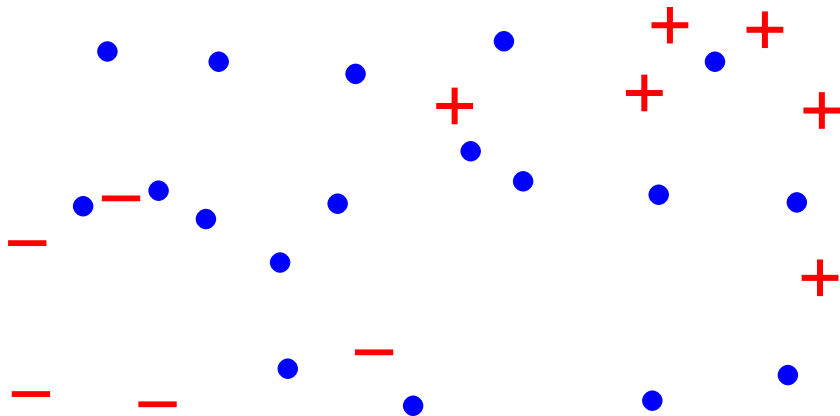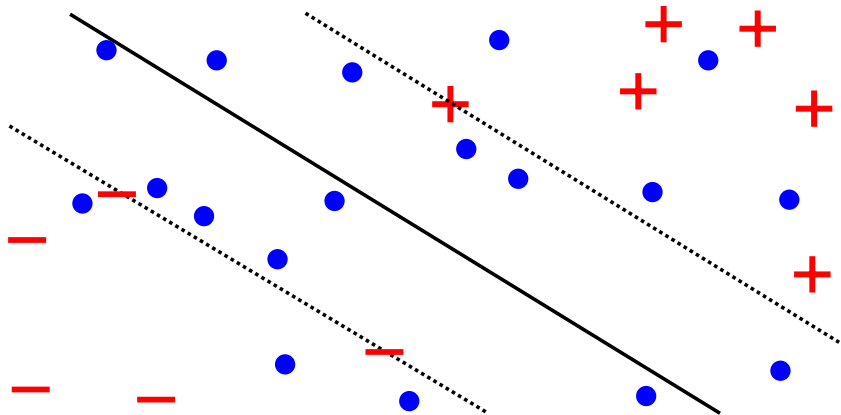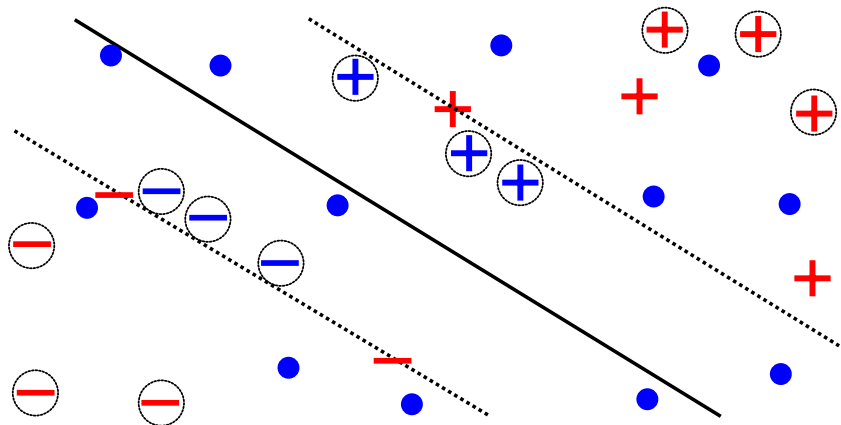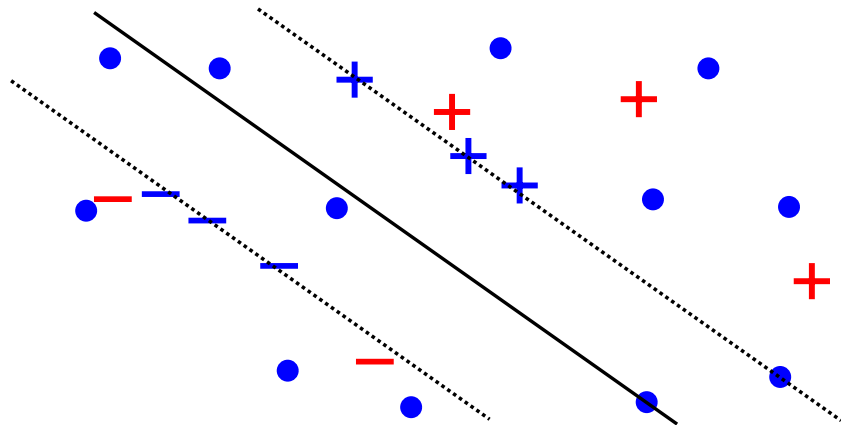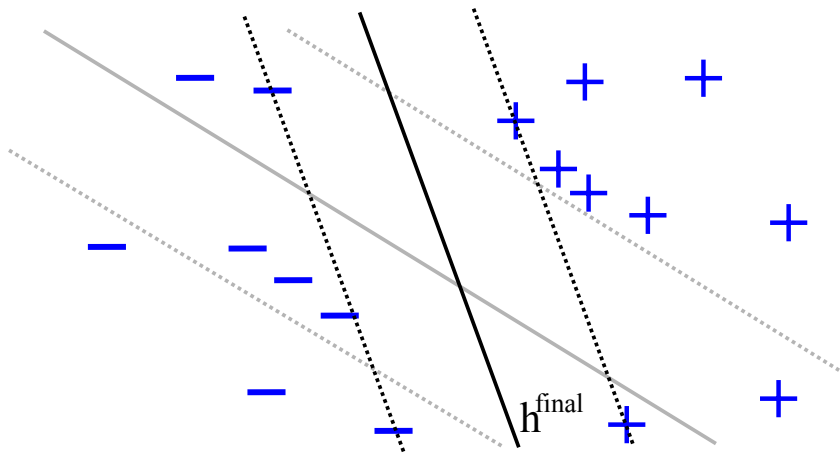
Convergence - theoretical guarantees

- What we need?: At each step pseudo-labels on target are sufficiently reasonable.
- $\Rightarrow$ Can be tackled with a notion of weak classifier

Weak classifier on a single domain

An hypothesis h^n, learned at iteration n, is a weak learner over a labeled sample S if it performs a bit better than a random guessing: $\exists \gamma_n \in]0; \frac{1}{2}$

$$R_S(h^n) = \hat{P}r_{(\mathbf{x}_i^s, y_i) \sim S^n}[h^n(\mathbf{x}_i) \neq y_i] = \frac{1}{2} - \gamma_n$$

A notion of weak learner for controlling pseudo-labels

Self labeling weak learner

A classifier $h^{(i)}$ learned at iteration over a current learning sample LS^i is self labeling weak learner w.r.t. a set SL^j of $2k$ pseudo-labeled examples introduced at step j if its true error over SL^j is strictly better than random guessing:

$$R_{SL^j}(h^{(i)}) = Pr_{\mathbf{x}_i^t \in SL^j}[h(\mathbf{x}_i^t) \neq y_i^t] < \frac{1}{2}$$

A first necessary condition

Theorem

Let $h^{(i)}$ a weak learner output at iteration i from $LS^{(i)}$, let
$\widetilde{R}_{LS^i}(h^{(i)}) = \frac{1}{2} - \gamma_{LS}^i$ the associated empirical error.
Let $R_T^{(i)} = \frac{1}{2} - \gamma_T^{(i)}$ the true empirical error over T.

$h^{(i)}$ is a self-labeling weak learner if $\gamma_{LS}^i > 0$

$\Rightarrow h^{(i)}$ will be able to correctly classify (w.r.t. their unknown true label) more than k pseudo-labeled target examples among $2k$ if at least half of them have been correctly pseudo-labeled.

A second result

Theorem

Let S a labeled source sample of m_S instances and T a target sample of $m_t \geq m_s$ unlabeled instances. Let $\mathcal{A}$ an iterative labeling algorithm using $2k$ examples at each step. $\mathcal{A}$ is able to perform an adaptation if

- $\gamma_S^{(i)} \geq \gamma_T^{(i)}, \forall i = 1...\frac{m_s}{2k}$

- $\gamma_S^{\max} > \sqrt{\frac{\gamma_T^{(0)}}{2}}$

$\Rightarrow h^{(i)}$ has to perform sufficiently well on the data it has been learned from
$\Rightarrow \mathcal{A}$ the final classifier has to work better on T than a classifier learned only from source data.

A simple illustration

Task: handwritten digit recognition. P_1 scaling problem between 1 and 0 and P_2 rotation problem between 5 and 7.

Iteration	P_1			P_2		
	$\gamma_S^{(i)}$	$\gamma_T^{(i-1)}$	$1 - \hat{\epsilon}_T^{(i)}$	$\gamma_S^{(i)}$	$\gamma_T^{(i-1)}$	$1 - \hat{\epsilon}_T^{(i)}$
1	0.5	0	0.585	0.50	-0.1	0.32
2	0.475	0.085	0.75	0.50	-0.18	0.285
3	0.48	0.25	0.73	0.50	-0.215	0.285
4	0.49	0.23	0.795	0.50	-0.215	0.24
5	0.49	0.295	0.875	0.50	-0.26	0.18
6	0.49	0.375	0.94	0.50	-0.32	0.205
7	0.49	0.44	0.94	0.50	-0.295	0.19
8	0.49	0.44	0.94	0.50	-0.31	0.12
9	0.49	0.44	0.94	0.50	-0.38	0.145
10	0.495	0.44	0.985	0.50	-0.355	0.115
11	0.5	0.485	0.99	0.495	-0.385	0.115

For P_1, we can check $\gamma_S^{(i)} \geq \gamma_T^{(i)}, \forall i = 1 \ldots \frac{m_s}{2k}$, and $\gamma_S^{\max} > \sqrt{\frac{\gamma_T^{(0)}}{2}}$.

- $h^{(i)}$ must work well on T
- $h^{(i)}$ must work well on S
- $\mathcal{A}$ works better than a non adaptation process

$\Rightarrow$ Necessary and reasonable conditions
$\Rightarrow$ Condition on T hard to check in practise
$\Rightarrow$ Boosting

Ensemble Methods and Boosting

Definition

Ensemble methods infer a set of classifiers $h_1, \ldots, h_N$ whose individual decisions are combined in some way to classify new examples.

Necessary conditions for an ensemble method to be efficient

- the individual classifiers are better than random guessing.
- they are diverse, *i.e.* they make different errors on new data points.

ADABOOST

- Learns step by step **weak** binary classifiers.
- **Optimizes a convex loss** by increasing the weights of misclassified examples.
- Builds a **convex combination** of the weak classifiers.

ADABOOST

Data: A learning sample S, a number of iterations N, a weak learner L
Result: A global hypothesis H_N
for $i = 1$ to m do $D_1(\mathbf{x_i}) = 1/m$;
for $t = 1$ to T do
$\quad h_n = \mathrm{LEARN}(S, \mathbf{D}_n)$;
$\quad \hat{\epsilon}_n = \sum_{\mathbf{x_i} \ s.t. \ y_i \neq h_n(\mathbf{x_i})} D_n(\mathbf{x_i})$;
$\quad \alpha_n = \frac{1}{2} \ln \frac{1 - \hat{\epsilon}_n}{\hat{\epsilon}_n}$;
$\quad$ for $i = 1$ to m do
$\quad\quad D_{t+1}(\mathbf{x_i}) = D_n(\mathbf{x_i}) \exp\left(-\alpha_t y_i h_t(\mathbf{x_i})\right) / Z_n$;
$\quad\quad$ /* Z_n is a normalization coefficient*/
$\quad$ end
end
$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_n h_n(\mathbf{x})$;
$H_N(\mathbf{x}) = \mathrm{sign}\left(f(\mathbf{x})\right)$;

Theoretical result on the empirical error

Theorem

Upper bound on the empirical error of the final classifier H_N

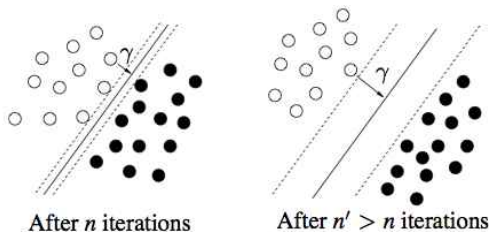$$\hat{\epsilon}_{H_N} \leq \prod_{t=1}^{T} Z_n \leq exp(-2 \sum_{t=1}^{T} \gamma_n^2)$$

where $\hat{\epsilon}_n = \frac{1}{2} - \gamma_n$ (weak hypothesis).
$\hat{\epsilon}_{H_N}$ is optimized with $\alpha_n = \frac{1}{2} \ln \frac{1-\hat{\epsilon}_n}{\hat{\epsilon}_n}$.

Meaning

This theorem means that the empirical error **exponentially decreases towards 0** with the number T of iterations.

Explanation in terms of margins of the training examples

After n iterations   After $n' > n$ iterations

Theorem

$\forall \gamma > 0$, with probability $1 - \delta$, any classifier ensemble H_T satisfies:

$$\epsilon_{H_T} \leq \mathbb{E}_{\mathbf{x} \in S}[margin(\mathbf{x}) \leq \gamma] + \mathcal{O}\left(\sqrt{\frac{d_h}{m} \frac{log^2(m/d_h)}{\gamma^2} + log(1/\delta)} \right),$$

where $\mathbb{E}_{\mathbf{x} \in S}[margin(\mathbf{x}) \leq \gamma]$ exponentially decreases towards 0 with T.

$\Rightarrow$ apply this idea on source and target (pseudo-labels)

Idea for domain adaptation

Double weighting strategy

- Keep the same weak hypothesis for both domains $h_1, \ldots, h_n, \ldots, h_N$
- Learn two functions
  - Source domain: $F_S^N = \sum_{n=1}^N \alpha_n h_n(\mathbf{x})$
  - Target domain: $F_T^N = \sum_{n=1}^N \beta_n h_n(\mathbf{x})$
- β_n depends on the (pseudo-)margin of the examples and a divergence measure
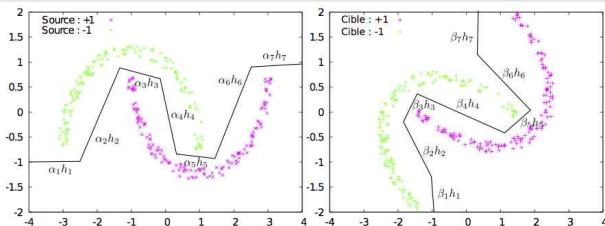
A notion of weak DA learner

Weak DA learner

A classifier h_n learned at iteration n from a S and T and a divergence $g_n \in [0, 1]$ between S and T and $f_{DA}(h_n(\mathbf{x}_i)) = |h_n(\mathbf{x}_i)| - \lambda g_n$, is a weak DA learner for T if:

1. h_n is a weak learner for S.

2. $\hat{L}_n = \mathbb{E}_{\mathbf{x}_i \sim T}[|f_{DA}(h_n(\mathbf{x}_i))| \leq \gamma] < \frac{\gamma}{\gamma + \max(\gamma, \lambda g_n)}$

- f_{DA}: obtaining high margin with small divergence

- if $\max(\gamma, \lambda g_n) = \gamma$: divergence is small, we are close to a semi-supervised setting

- if $\max(\gamma, \lambda g_n) = \lambda g_n$: divergence is high and the reweighting scheme requires a specific attention to the divergence.

Algorithm SLDAB

SLDAB

Data: Learning sample S, Nb of iterations N, unlabeled sample T, $\gamma \in [0,1]$, $\lambda \in [0,1]$
Result: Source and target hypothesis H_S, H_T
foreach $\forall (\mathbf{x}_i^s, y_i^s) \in S$, $\mathbf{x}_j^t \in T$ **do** $D_1^S(\mathbf{x}_i^s) = 1/m_s$; $D_1^T(\mathbf{x}_j^t) = 1/m_t$;
for $n = 1$ **to** N **do**

    Learn h_n to produce a **weak DA** learner; compute g_n;

    $\alpha_n = \frac{1}{2} \ln \frac{1 - \hat{\epsilon}_{Sn}(h^n)}{\hat{\epsilon}_{Sn}(h^n)}$; $\beta_n = \frac{1}{\gamma + \max(\gamma, \lambda g_n)} \ln \frac{\gamma W_n^+}{\max(\gamma, \lambda g_n) W_n^-}$;

    for $(\mathbf{x}_i^s, y_i^s) \in S$ **do** $D_{n+1}(\mathbf{x}_i) = D_n(\mathbf{x}_i^s) \exp\left(-\alpha_n y_i^s \text{sign}(h_n(\mathbf{x}_i^s))\right) / Z_S^n$;

    for $\mathbf{x}_j^t \in T$ **do**

        $D_T^{n+1}(\mathbf{x}_j) = D_T^n(\mathbf{x}_j^t) \exp\left(-\beta_n y_j^n f_{DA}(h_n(\mathbf{x}_j^t))\right) / Z_T^n$;

        where $y_j^n = -\text{sign}(f_{DA}(h_n(\mathbf{x})))$ if $|f_{DA}(h_n(\mathbf{x}))| > \gamma$ and $y_j^n = -\text{sign}(f_{DA}(h_n(\mathbf{x})))$
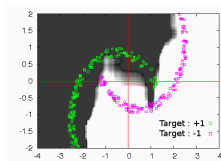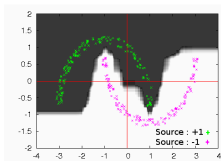
        otherwise;

    end

end

$F_S^N(\mathbf{x}^s) = \sum_{n=1}^{N} \alpha_n \text{sign}(h_n(\mathbf{x}^s))$;
$H_T^N(\mathbf{x}^t) = \sum_{n=1}^{N} \beta_n \text{sign}(h_n(\mathbf{x}^t))$;

Conclusion

Theoretical results

- Convergence of the empirical losses (exponentially fast to 0) $\Rightarrow$ (pseudo-)margin increases
- No generalization bound over the true error on P_T

Difficult aspects

- Defining divergence g_n: avoid degenerate cases
- Finding/Learning weak DA learned: need to take into account both source error and divergence information
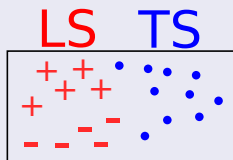
Model selection?

Reverse validation [Zhong et aL.,ECML'10;Bruzzone et al.,PAMI'10]

Reverse classifier h^r
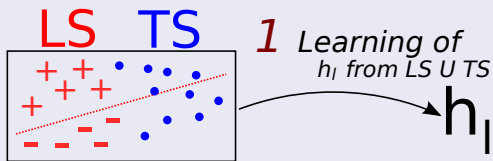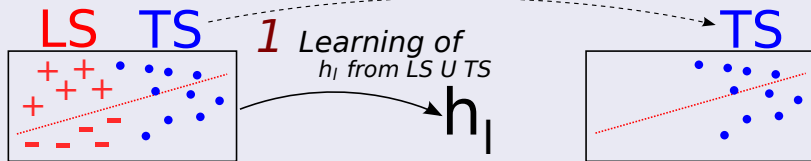
Reverse validation [Zhong et aL.,ECML'10;Bruzzone et al.,PAMI'10]

Reverse classifier h^r

Reverse validation [Zhong et aL.,ECML'10;Bruzzone et al.,PAMI'10]

Reverse classifier h^r

LS TS **1** *Learning of h_l from LS ∪ TS*

h_l

Reverse validation [Zhong et aL.,ECML'10;Bruzzone et al.,PAMI'10]

Reverse validation [Zhong et aL.,ECML'10;Bruzzone et al.,PAMI'10]

Reverse validation [Zhong et aL.,ECML'10;Bruzzone et al.,PAMI'10]

Reverse classifier h^r

LS TS *1* *Learning of h_l from LS U TS* h_l

2 *Auto Labeling of TS with h_l* TS

h_l^r *3* *Learning of h_l^r from TS auto labeled*

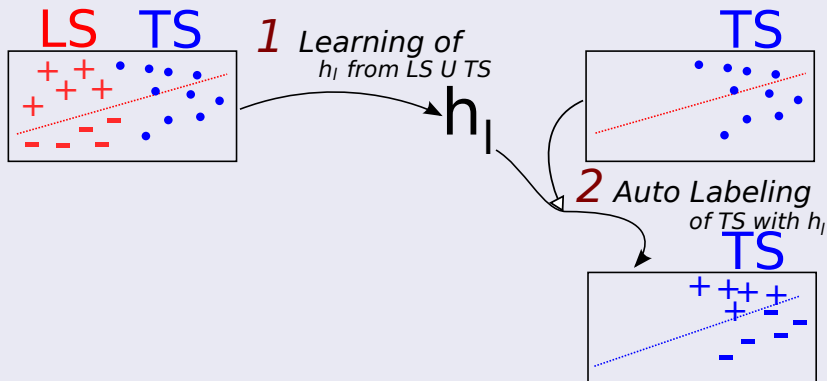# Reverse validation [Zhong et aL.,ECML'10;Bruzzone et al.,PAMI'10]

Reverse validation [Zhong et aL.,ECML'10;Bruzzone et al.,PAMI'10]

Reverse classifier h^r

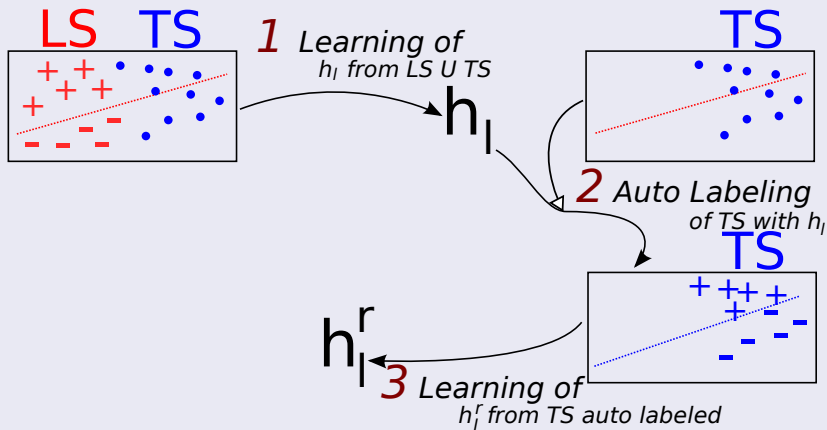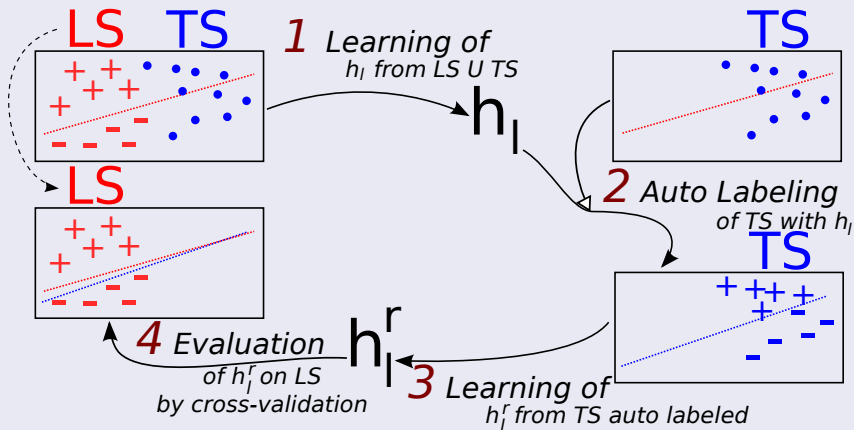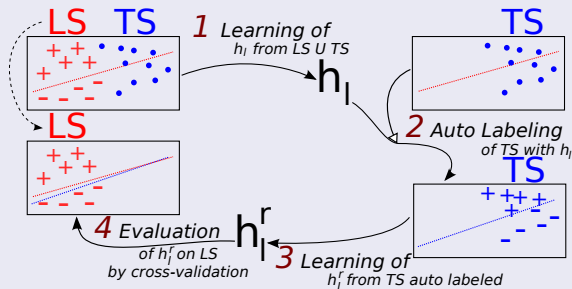- Two domains are related $\Rightarrow h_l^r$ performs well on the source domain
- Used with target labels to have an estimation of R_{P_T}
- Used to heuristically estimate theoretical constants of adaptability (λ) [Morvant et al.,ICDM'11;KAIS'12]

Conclusion

Conclusion

- Very active domains - Lots of methods (Sometimes difficult to follow) Approaches not covered here: probabilistic-based, bayesian, deep learning methods, etc.
- Same idea: Moving closer the distributions while ensuring good accuracy on labeled data
- Can we imagine general efficient frameworks
  $\Rightarrow$ probably No: DA is difficult [Ben-David et al.,ALT'12]
  $\Rightarrow$ Choose a method in function of the task/data
- Importance of data preparation
- Importance of divergence measures

Perspectives

- Understanding negative transfer
- Model selection
- Heterogeneous data
- Large scale
- Links with multi-tasks and multi-source learning, lifelong learning, concept drift, etc.
- $\Rightarrow$ A large room for more research

- J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schlkopf, "Correcting sample selection bias by unlabeled data,", NIPS 2007.
- S. J. Pan and Q. Yang, "A survey on transfer learning", TKDE 2010.
- H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function", Journal of Statistical Planning and Inference, 2000.
- H. Daumé III, "Frustratingly easy domain adaptation", ACL, 2007.
- H. Daumé III, A. Kumar, et A. Saha. Co-regularization based semi-supervised domain adaptation. NIPS 2010.
- R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach", ICCV'11.
- B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation", CVPR 2012.
- K. Saenko, B. Kulis, M. Fritz, and T. Darrell., "Adapting visual category models to new domains", ECCV'2010.
- J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning", ICML 2007.

References (partial) (2/5)

- B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms", CVPR'11

- B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars,"Unsupervised visual domain adaptation using subspace alignment", ICCV 2013

- L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy", PAMI 2010.

- B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation", ICML 2013

- X. Glorot, A. Bordes, and Y. Bengio, "Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach", ICML 2011.

- P. Germain, A. Habrard, F. Laviolette, et E. Morvant. "PAC-Bayesian domain adaptation bound with specialization to linear classifiers", ICML 2013.

- C. Cortes et M. Mohri. Domain adaptation in regression. ALT 2011.

- S. Ben-David, J. Blitzer, K. Crammer, et F. Pereira. Analysis of representations for domain adaptation. NIPS 2006.

References (partial) (3/5)

- S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, et J.W. Vaughan. A theory of learning from different domains. Machine Learning 2010.
- S. Ben-David et R. Urner. On the hardness of domain adaptation and the utility of unlabeled target samples. ALT 2012.
- S. Ben-David et R. Urner. On the hardness of domain adaptation and the utility of unlabeled target samples. AISTAT 2010.
- J. Ye C. Zhang, L. Zhang. Generalization bounds for domain adaptation. NIPS 2012.
- M. Chen, K. Q. Weinberger, et J. Blitzer. Co-training for domain adaptation. NIPS 2011.
- A. Habrard, JP Peyrache, M. Sebban. Iterative Self-labeling Domain Adaptation for Linear Structured Image Classification, IJAIT 2013.
- A. Habrard, JP Peyrache, M. Sebban. Boosting for unsupervised domain adaptation, ECML 2013
- Y. Mansour, M. Mohri, et A. Rostamizadeh. Domain adap- tation : Learning bounds and algorithms. COLT 2009.
- Y. Mansour, M. Mohri, et A. Rostamizadeh. Multiple source adaptation and the rényi divergence. UAI 2009.

References (partial) (4/5)

- M. Mohri et A.M. Medina. New analysis and algorithm for learning with drifting distributions. ALT 2012.
- E. Morvant, A. Habrard, et S. Ayache. Sparse domain adaptation in projection spaces based on good similarity functions. ICDM 2011.
- E. Morvant, A. Habrard, et S. Ayache. Parsimonious Unsupervised and Semi-Supervised Domain Adaptation with Good Similarity Functions. KAIS 2012.
- E. Zhong, W. Fan, Q. Yang, O. Verscheure, et J. Ren. Cross validation framework to choose amongst models and datasets for transfer learning. ECML 2010.
- C. Cortes and M. Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. Theoretical Computer Science, 2013.
- M. Sugiyama, S. Nakajima, H. Kashima, P. von Bunau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. NIPS 2008.
- C. Cortes, M. Mohri, A. Medina. Adaptation Algorithm and Theory Based on Generalized Discrepancy, 2014.
- S. Bickel, M. Bruckner, and T. Scheffer, "Discriminative learning for differing training and test distributions", 2007

- B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation", CVPR 2012
- A. Torralba and A. A. Efros, "Unbiased look at dataset bias", CVPR 2011.
- A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, B. Scholkopf. Covariate Shift by Kernel Mean Matching. 2008.