Theoretical Analysis of Domain Adaptation
Current state of the art

Shai Ben-David

September 14, 2012

Domain Adaptation

Most of the statistical learning guarantees are based on assuming that *the learning environment is unchanged throughout the learning process.*

Domain Adaptation

Most of the statistical learning guarantees are based on assuming that *the learning environment is unchanged throughout the learning process.*

Formally, it is common to assume that *both the training and the test examples are generated i.i.d. by the same fixed probability distribution.*

Domain Adaptation

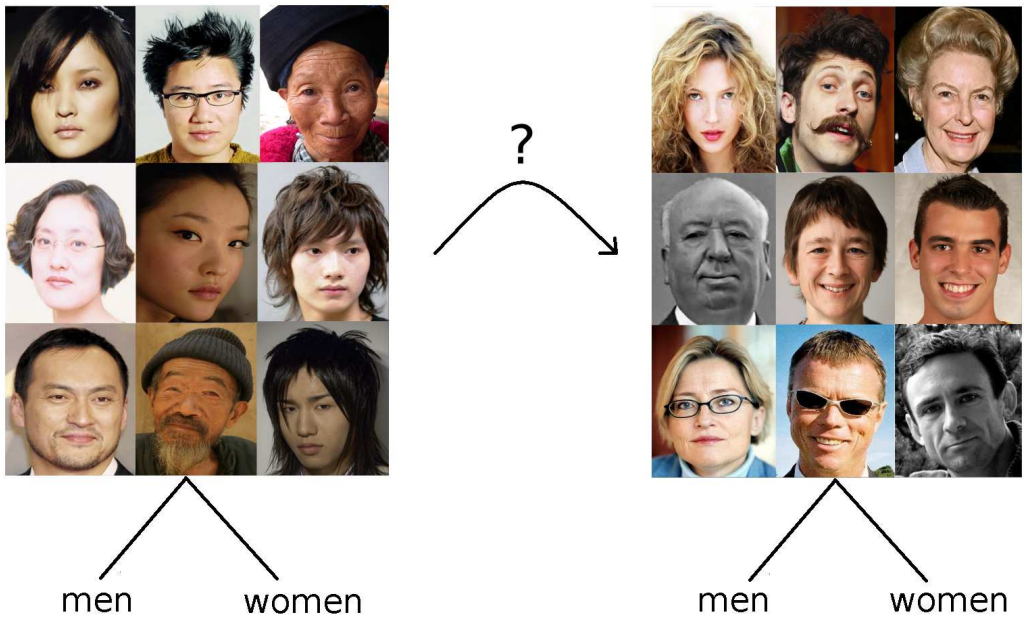
Most of the statistical learning guarantees are based on assuming that *the learning environment is unchanged throughout the learning process.*

Formally, it is common to assume that *both the training and the test examples are generated i.i.d. by the same fixed probability distribution.*

This is unrealistic for many ML applications

Learning when Training and Test distributions differ

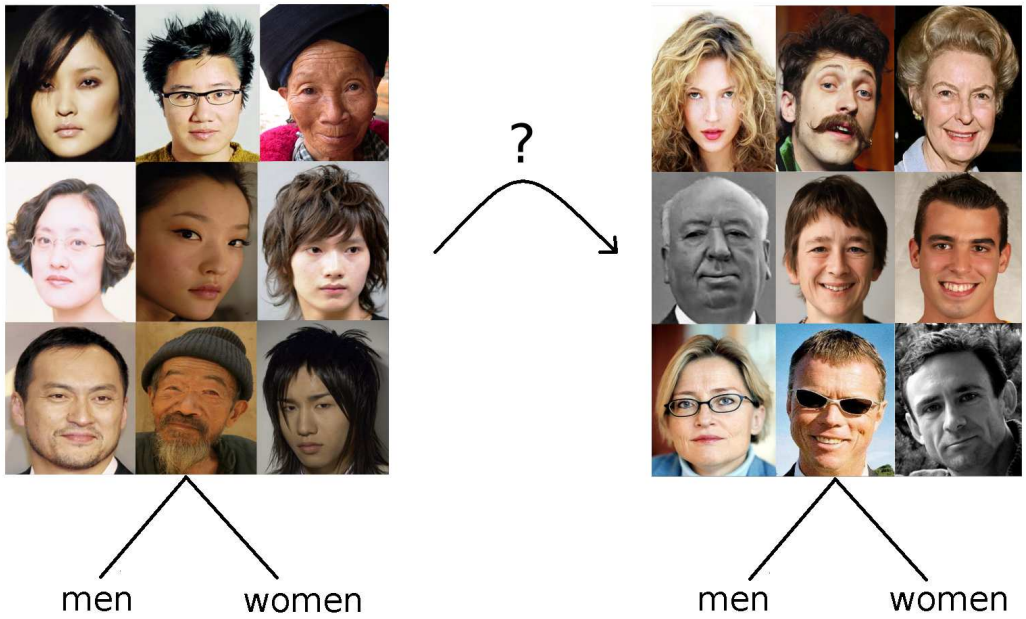
Examples:



- Spam filters – train on email arriving at one address, test on a different mailbox.
- Natural Language Processing tasks- train on some content domains, test on others.

Learning when Training and Test distributions differ

Examples:



- Spam filters – train on email arriving at one address, test on a different mailbox.
- Natural Language Processing tasks- train on some content domains, test on others.

There is rather little theoretical understanding so far.

Why care about theoretical understanding?

- Know when to use (and when not to use) algorithmic paradigms.

Why care about theoretical understanding?

- Know when to use (and when not to use) algorithmic paradigms.
- Have some performance guarantees.

Why care about theoretical understanding?

- Know when to use (and when not to use) algorithmic paradigms.
- Have some performance guarantees.
- Help choose appropriate algorithmic approach (based on prior knowledge about the task at hand).

Why care about theoretical understanding?

- Know when to use (and when not to use) algorithmic paradigms.
- Have some performance guarantees.
- Help choose appropriate algorithmic approach (based on prior knowledge about the task at hand).
- The joy of understanding .....

Example: Domain adaptation for POS tagging

Structural Correspondence Learning(Blitzer, McDonald, Pereira 2005):

1. Choose a set of *pivot words* (determiners, prepositions, connectors and frequently occurring verbs).
2. Represent every word in a text as a vector of its correlations with each of the pivot words.
3. Train a linear separator on the (images of) the training data coming from one domain and use it for tagging on the other.

Abstraction and analysis (BD, Blitzer, Crammer, Pereira 2005)

- Embed the original attribute space into some joint feature space in which:
  1. The two tasks look similar.
  2. The source task can still be well classified.

Abstraction and analysis (BD, Blitzer, Crammer, Pereira 2005)

- Embed the original attribute space into some joint feature space in which:
  1. The two tasks look similar.
  2. The source task can still be well classified.

- Then, treat the images of points from both distributions as if they are coming from a single distribution.

Formalism

Domain: $\mathcal{X}$
Label set: $\{0, 1\}$
Source Distribution: P_S over $\mathcal{X} \times \{0, 1\}$
Target Distribution: P_T over $\mathcal{X} \times \{0, 1\}$

A *DA-learner* gets a labeled sample S from the source and a (large) unlabeled sample T from the target and outputs a label predictor

$$h : \mathcal{X} \to \{0, 1\}.$$

Goal: Learn a predictor with small target error

$$\mathrm{Err}_{P_T}(h) := \Pr_{(x,y)\sim P_T}[h(x) \neq y] \leq \epsilon$$

The error bound supporting that paradigm

[BD, Blitzer, Crammer, Pereira 2006]
[Mansour, Mohri, Rostamizadeh 2009]
For all $h \in H$:

$$\mathrm{Err}^T(h) \leq \mathrm{Err}^S(h) + A + \lambda,$$

Where A is an additive measure of discrepancy between the marginals and λ a measure of the discrepancy between the labels, both depending on H.

The error bound supporting that paradigm

[BD, Blitzer, Crammer, Pereira 2006]
[Mansour, Mohri, Rostamizadeh 2009]
For all $h \in H$:

$$\mathrm{Err}^T(h) \leq \mathrm{Err}^S(h) + A + \lambda,$$

Where A is an additive measure of discrepancy between the marginals and λ a measure of the discrepancy between the labels, both depending on H.

Namely,
$A = d_{H\Delta H}(P_T, P_S) \stackrel{def}{=} Sup\{|P_T(h\Delta h') - P_S(h\Delta h')| : h, h' \in H\}$

The error bound supporting that paradigm

[BD, Blitzer, Crammer, Pereira 2006]
[Mansour, Mohri, Rostamizadeh 2009]
For all $h \in H$:

$$\mathrm{Err}^T(h) \leq \mathrm{Err}^S(h) + A + \lambda,$$

Where A is an additive measure of discrepancy between the marginals and λ a measure of the discrepancy between the labels, both depending on H.

Namely,
$A = d_{H\Delta H}(P_T, P_S) \stackrel{def}{=} Sup\{|P_T(h\Delta h') - P_S(h\Delta h')| : h, h' \in H\}$

and
$\lambda = Inf\{\mathrm{Err}^T(h) + \mathrm{Err}^S(h) : h \in H\}$
(The Mansour et al result uses a variation of this - $\mathrm{Err}^T(h_S) + \mathrm{Err}^S(h_T)$, where h_S and h_T are minimum error classifiers in H for P_S and P_T, respectively).

From the bound to an algorithm

The bounds imply error guarantees for any algorithm that learns well with respect to the source task.

From the bound to an algorithm

The bounds imply error guarantees for any algorithm that learns well with respect to the source task.

For example, the simple empirical risk minimization $ERM(H)$ paradigms,
provided that H has limited capacity (say, finite VC-dimension).

Overview

Three aspects determining a DA framework:

1. The type of training samples available to the learner.
2. The assumptions on the relationship between the source (training) and target (test) data-generating distributions.
3. The prior knowledge about the task that the learner has.

Overview

Three aspects determining a DA framework:

1. The type of training samples available to the learner.
2. The assumptions on the relationship between the source (training) and target (test) data-generating distributions.
3. The prior knowledge about the task that the learner has.

Two types of algorithms:

1. Conservative: Learn the source task and apply the result to the target.
2. Adaptive: Adapt the output classifier based on target information.

The training samples available to the learner

Types of "proxy data"

- labeled data from a different distribution (source distribution)
- (lots of) unlabeled data from the target distribution

The training samples available to the learner

Types of "proxy data"

- labeled data from a different distribution (source distribution)
- (lots of) unlabeled data from the target distribution

Questions:

- Can we learn with solely with source generated labeled data?
- Can target-generated unlabeled data be beneficial or even necessary?
- How can we utilize the proxy data if we are also given (little) labeled data from the target distribution?

Relatedness assumptions

Relatedness of the unlabeled marginal distributions

- Multiplicative measure of distance (the *ratio* between the source and target probabilities of domain subsets).
- Additive measure of distance (the *difference* between the source and target probabilities of domain subsets, like the $d_{H\Delta H}$ above)

(both with respect to some family of domain subsets)

Relatedness of the labeling functions

- Absolute (like the *covariate shift* assumption)
- Relative to a hypothesis class (like the λ parameter above)

Prior knowledge

Prior knowledge about either the source task or the target task.
For example:

- Realizability by some class of predictors.
- Good approximation by some class
- Good kernel

Prior knowledge

Prior knowledge about either the source task or the target task.
For example:

- Realizability by some class of predictors.
- Good approximation by some class
- Good kernel

What are the differences between source and target prior knowledge?

The downside of conservative algorithms

They can thus be viewed as indicating

"When is domain adaptation not needed?"

(the algorithm is just learning with respect to the source-generated traing data)

Adaptive algorithms:

A common adaptive paradigm is importance reweighing.

Namely, reweigh the source-generate labeled training sample, such that it will look as if it was generated by the target task.

Adaptive algorithms:

A common adaptive paradigm is importance reweighing.

Namely, reweigh the source-generate labeled training sample, such that it will look as if it was generated by the target task.

This is a rather common paradigm in practice.

Adaptive algorithms:

A common adaptive paradigm is importance reweighing.

Namely, reweigh the source-generate labeled training sample, such that it will look as if it was generated by the target task.

This is a rather common paradigm in practice.

However, for a theoretical justification of this paradigm, we need some further assumptions.

Relatedness assumptions for the labeling: Covariate shift

The covariate- shift assumption: The labeling function is the same for the source and target tasks.

(This is reasonable for some DA tasks, such as parts of speech tagging, but may fail in others).

Relatedness assumptions for the amrginals: Weight Ratio

We define the weight ratio of the source distribution and the target distribution with respect to some collection of subsets $\mathcal{B} \subseteq 2^{\mathcal{X}}$ as

$$C_{\mathcal{B}}(D_S, D_T) = \inf_{\substack{b \in \mathcal{B} \\ D_T(b) \neq 0}} \frac{D_S(b)}{D_T(b)}$$

Rational: We want the source domain to be not-too-sparse in areas that are dense from the target's perspective.

An observation using a point-wise weight ratio assumption

If $C_{\{\{x\}|x\in\mathcal{X}\}}(P_S, P_T) > 0$, we have for every $h \in \{0,1\}^{\mathcal{X}}$

$$\mathrm{Err}_T(h) \;\leq\; \frac{1}{C_{\{\{x\}|x\in\mathcal{X}\}}(P_S, P_T)}\mathrm{Err}_S(h)\ .$$

An observation using a point-wise weight ratio assumption

If $C_{\{\{x\}|x\in\mathcal{X}\}}(P_S, P_T) > 0$, we have for every $h \in \{0,1\}^{\mathcal{X}}$

$$\mathrm{Err}_T(h) \;\leq\; \frac{1}{C_{\{\{x\}|x\in\mathcal{X}\}}(P_S, P_T)}\mathrm{Err}_S(h)\ .$$

Thus, any algorithm that (ϵ, δ)-learns the source for arbitrarily small ϵ and δ also learns the target.

An observation using a point-wise weight ratio assumption

If $C_{\{\{x\}|x\in\mathcal{X}\}}(P_S, P_T) > 0$, we have for every $h \in \{0,1\}^{\mathcal{X}}$

$$\mathrm{Err}_T(h) \;\leq\; \frac{1}{C_{\{\{x\}|x\in\mathcal{X}\}}(P_S, P_T)}\mathrm{Err}_S(h)\ .$$

Thus, any algorithm that (ϵ, δ)-learns the source for arbitrarily small ϵ and δ also learns the target.

No unlabeled target data needed (if one ignores the issue of sample sizes).

A first drawback of the point-wise weight ratio assumption

The result may become meaningless if there is a non-zero lower bound on the error achievable (e.g. when the labeling rule is not deterministic or due to non-zero approximation error of the class of predictors that the algorithm considers).

Adaptive algorithms under the point-wise weight ratio assumption

[Cortes, Mansour, Mohri 2010] prove guarantees for a paradigm reweighing the training data based on covariate shift and knowledge of the point-wise density ratio between source and target.

Adaptive algorithms under the point-wise weight ratio assumption

[Cortes, Mansour, Mohri 2010] prove guarantees for a paradigm reweighing the training data based on covariate shift and knowledge of the point-wise density ratio between source and target.

[BD, Lu, Luu, Pal 2010] show necessity of the assumptions for these results.

A second drawback of the point-wise weight ratio assumption

A bound on the point-wise weight ratio is a rather strong assumption..

A second drawback of the point-wise weight ratio assumption

A bound on the point-wise weight ratio is a rather strong assumption..

Furthermore, [BD, Urner 2012] show that *learning* that point-wise density ratio may require unrealistically large target-generated samples.

A second drawback of the point-wise weight ratio assumption

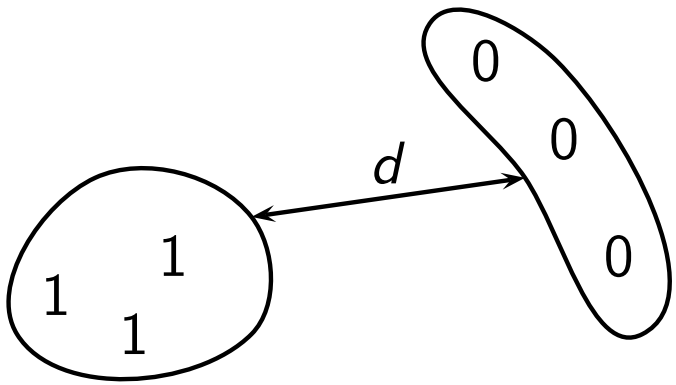
A bound on the point-wise weight ratio is a rather strong assumption..

Furthermore, [BD, Urner 2012] show that *learning* that point-wise density ratio may require unrealistically large target-generated samples.

However under a new Lipschitzness assumption, this assumption can be relaxed.

Lipschitzness of the labeling rule

The labeling rule satisfies a Lipschitzness assumption (if and) only if the data splits into well-separated label-homogenous clusters,.



Lipschitz condition:
$|l(x) - l(y)| \leq 1/d \, \|x - y\|$

Assuming that natural learning tasks have such a property is unrealistically optimistic.

A new property - Probabilistic Lipschitzness ([Urner, Shalev-Shwartz, BD, 2011])

Definition

Let $\phi : \mathbb{R} \to \mathbb{R}$. We say that $h : \mathcal{X} \to \mathbb{R}$ is ϕ-Lipschitz w.r.t. a distribution D over $\mathcal{X}$ if the following holds for all $\lambda > 0$:

$$P_{x \sim D}[\exists y : |h(x) - h(y)| > \lambda \|x - y\|] \leq \phi(\lambda)$$

Essentially, the condition asserts that the boundaries between class-labels go through sparsely populated domain regions.

This may be viewed as a formalization of the, often loosely stated, *cluster assumption*.

Nearest Neighbor for domain adaptation [BD, Urner, Shalev-Swartz 2012]

Algorithm:

Given a labeled sample S from the source, label each test point t from the target by its nearest neighbor in S.

Nearest Neighbor for domain adaptation [BD, Urner, Shalev-Swartz 2012]

Algorithm:

Given a labeled sample S from the source, label each test point t from the target by its nearest neighbor in S.

We provide finite sample size error guarantees for this algorithm under our assumptions.

Nearest-Neighbor learning guarantee

Theorem

Let our domain $\mathcal{X} = [0,1]^d$ be the unit cube in $\mathbb{R}^d$ and let $\mathcal{W}$ be the class of pairs (P_S, P_T) of source and target distributions over $\mathcal{X} \times \{0,1\}$ with $C_B(D_S, D_T) = C > 0$ satisfying the covariate shift assumption and their common labeling function $l : \mathcal{X} \to [0,1]$ satisfying the ϕ-probabilistic-Lipschitz property. Then, for all λ we have

$$\mathbb{E}_{S \sim P_S^m}[\mathrm{Err}_{P_T}(h_{\mathrm{NN}})] \leq 2\mathrm{opt}(P_T) + \phi(\lambda) + 4\lambda \frac{\sqrt{d}}{C} \left(\frac{1}{m}\right)^{\frac{1}{d+1}} .$$

Is the dependence on the Lipschitness necessary?

[BD, Urner 2012], show lower bounds on the needed training sample sizes.

In particular, without the L assumption, any algorithm requires sample sizes of the order of the full domain size! Details below.

The prior knowledge about the task that the learner has

The third aspect determining a DA problem is the nature of the prior knowledge available to the earner.

We consider two such scenarios:

1. The learner knows some class of predictors, H_S that has zero approximation error w.r.t. the source data distribution.

2. The learner knows some class of predictors, H_T that has zero approximation error w.r.t. the target data distribution.

DA with learner's prior knowledge

We show that in the first case, learning is possible without use of unlabeled target-generated samples.

DA with learner's prior knowledge

We show that in the first case, learning is possible without use of unlabeled target-generated samples.

However, in the second scenario, there are provable benefits to using (very large) unlabeled target-generated samples.

Source realizability

Knowing a class H_S of finite VC-dimension that realizes the source implies that ERM(H_S) is a successful learning paradigm for the source distribution that achieves arbitrarily small error.

In such a case, empirical risk minimization w.r.t. the source-generated training sample yields arbitrarily PAC learning, and, as mentioned above, if the point wise weight-ratio, $C(P_S, P_T$, is non-zero, such an algorithm will also yield PAC learning of the target task.

Source realizability

Knowing a class H_S of finite VC-dimension that realizes the source implies that ERM(H_S) is a successful learning paradigm for the source distribution that achieves arbitrarily small error.

In such a case, empirical risk minimization w.r.t. the source-generated training sample yields arbitrarily PAC learning, and, as mentioned above, if the point wise weight-ratio, $C(P_S, P_T$, is non-zero, such an algorithm will also yield PAC learning of the target task.

No target data is needed. (learning is possible from just source-generated samples who's sizes are only a constant times the sizes needed for learning the source task).

A lower bound under target realizability [BD, Urner 2012]

Assume the learner knows a class H_T that realizes the target distribution.

Theorem

For every finite domain $\mathcal{X}$ there exists a class H_T with $\mathrm{VCdim}(H_T) = 1$ such that for every ϵ and δ with $\epsilon + \delta < 1/2$, no algorithm can (ϵ, δ, s, t)-solve the realizable DA problem for the class $\mathcal{W}$ of triples (P_S, P_T, l) with $C(P_S, P_T) \geq 1/2$ and $\mathrm{opt}^l_T(H_T) = 0$ if $s + t < \sqrt{(1 - 2(\epsilon + \delta))|\mathcal{X}|}$.

A lower bound under target realizability [BD, Urner 2012]

Assume the learner knows a class H_T that realizes the target distribution.

Theorem

For every finite domain $\mathcal{X}$ there exists a class H_T with $\mathrm{VCdim}(H_T) = 1$ such that for every ϵ and δ with $\epsilon + \delta < 1/2$, no algorithm can (ϵ, δ, s, t)-solve the realizable DA problem for the class $\mathcal{W}$ of triples (P_S, P_T, l) with $C(P_S, P_T) \geq 1/2$ and $\mathrm{opt}^l_T(H_T) = 0$ if $s + t < \sqrt{(1 - 2(\epsilon + \delta))|\mathcal{X}|}$.

In other words, this assumption will not suffice to guarantee meaningful learning with samples that are independent of the domain size.

A lower bound under target realizability [BD, Urner 2012]

Assume the learner knows a class H_T that realizes the target distribution.

Theorem

For every finite domain $\mathcal{X}$ there exists a class H_T with $\mathrm{VCdim}(H_T) = 1$ such that for every ϵ and δ with $\epsilon + \delta < 1/2$, no algorithm can (ϵ, δ, s, t)-solve the realizable DA problem for the class $\mathcal{W}$ of triples (P_S, P_T, l) with $C(P_S, P_T) \geq 1/2$ and $\mathrm{opt}^l_T(H_T) = 0$ if $s + t < \sqrt{(1 - 2(\epsilon + \delta))|\mathcal{X}|}$.

In other words, this assumption will not suffice to guarantee meaningful learning with samples that are independent of the domain size.

[BD, Urner 2012] also show a almost-matching upper bound on the needed sample sizes.

Is unlabeled target data unnecessary?

Does a (point-wise) weight ratio assumption always allow to replace a target generated labeled sample solely by (possibly lots of) source generated labeled examples?

Is unlabeled target data unnecessary?

Does a (point-wise) weight ratio assumption always allow to replace a target generated labeled sample solely by (possibly lots of) source generated labeled examples?

Answer: No! There are situations, where even under these strong assumptions, target-generated data is provably necessary for successful learning.

Proper DA-learning [BD, Urner, Shalev-Swartz 2012]

Sometimes we want to learn a classifier from a specific class , e.g.

- a class of fast classifiers
- a class of functions that are readily interpretable

(e.g. halfspaces or small decision trees)

Proper DA-learning [BD, Urner, Shalev-Swartz 2012]

Sometimes we want to learn a classifier from a specific class , e.g.

- a class of fast classifiers
- a class of functions that are readily interpretable

(e.g. halfspaces or small decision trees)

Problem:

Given	A hypothesis class H of interest
Input	Source sample S and unlabeled target sample T
Output	A member of the class H with low error

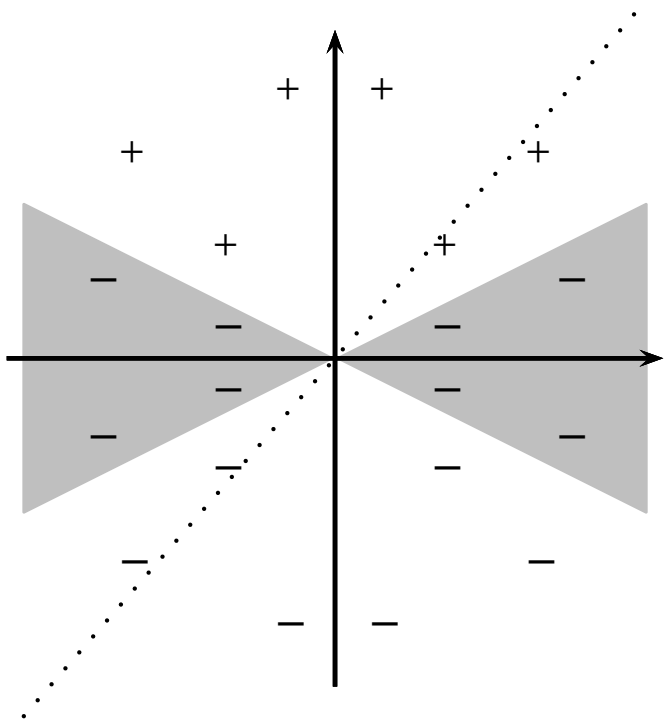
Algorithm

Algorithm:

- Use a DA-learner to learn a labeling function f for P_T
- Use f to label an unlabeled sample T from the target
- Feed T to an agnostic learner to an agnostic learner for H

This algorithm DA-learns H.

Benefit of unlabeled data



Domain: Unit cube

Source: Uniform

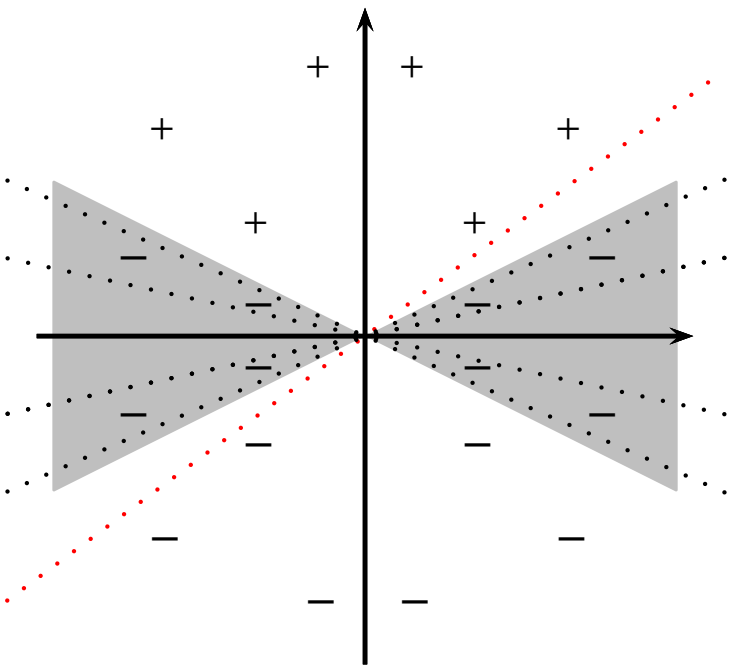
Target: Support in grey area

Labeling: As in image

Class H: Homogeneous halfspaces

Weight ratio: $C > 0$

Benefit of unlabeled data



For the source, many classifiers are equally good/bad.

Thus it becomes crucial to estimate, which half of the grey area is heavier according to the target distribution.

This can not be done without data generated by the target.

We can first label the unlabeled target-data with a nearest neighbor algorithm and then feed this labeled target data to an H-learner.

Summary

We investigated which assumptions allow which kind of replacement of "perfect" data by "proxy" data:

- For some algorithms labeled source data suffices:
  - Learners that achieve arbitrary small error on the source (source realizability)
  - Nearest Neighbor
- There are scenarios where (unlabeled) target data is provably necessary and beneficial:
  - Proper DA-learning.
  - When there is prior knowledge about a class of predictors that do well on the target task.

Many open questions

- Which assumptions make sense in practice?
- Are there adaptive algorithms that can guaranteed to succeed based on realistic assumptions?
- Analyze the utility of (relatively few) labeled target-generated examples.