Utilizing Unlabeled Data for Classification-Prediction Learning

Shai Ben-David,
Ruth Urner

Shai Shalev-Shwartz,
Ohad Shamir,
and Shalev Ben-David

In many applications unlabeled data is cheap and abundant

- Classification of emails for spam detection.
- Parts of speech tagging of text.
- Classifying tweets by their sentiment.

The issues we wish to address

- When can unlabeled training data be utilized to help classification learning?
- Algorithmic paradigms for utilizing unlabeled (and "weakly labeled") data.
- Theoretical analysis of the benefits of such data.

What's in this talk?

I'll describe three paradigms for utilizing unlabeled samples (ULS) for classification learning tasks.

1. **"Proper" Semi Supervised Learning** – Using ULS to choose the best classifier from a given set of desirable classifiers.
2. **Domain Adaptation** – using target ULS to reweigh available source-generated training samples.
3. **Learning for weak teachers** – How can, say, *Amazon's Mechanical Turk* supervision be utilized to save correctly labeled data?

Using unlabeled data to improve prediction accuracy

Intuitively, this may help if

1. The underlying data distribution is nicely clusterable.
2. The clusters correlate with the labels. Namely, the clusters are (roughly) homogeneously labeled.

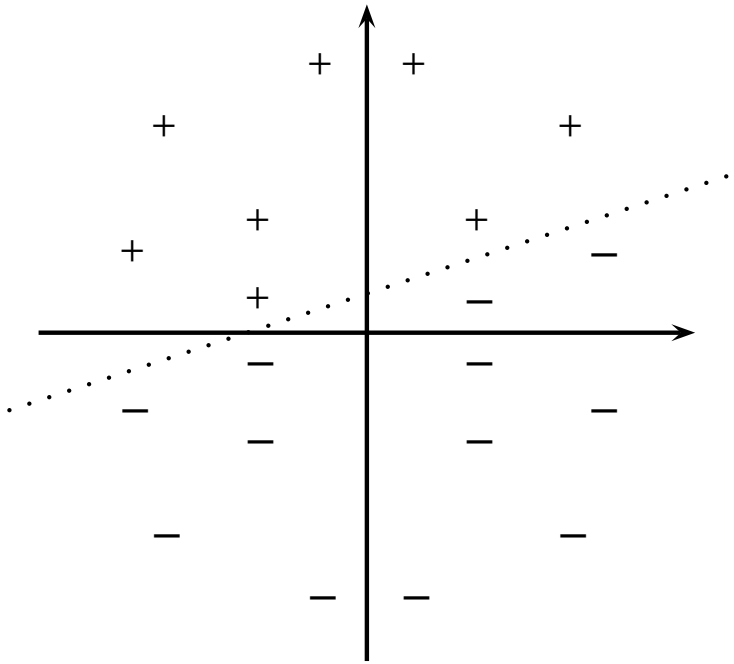
There are some issues with the above intuition;

1. It is not clear how to phrase the first assumption in a formal way that is still practically relevant.
2. Without the second assumption, it can be proven that unlabeled data has very limited utility [BD, Lu, Luu, Pal, 2010]

“Proper Learning” - Desirable classifiers

Applications pose specific requirements on classifiers, such as being

- fast at prediction time
- readily interpretable



Linear halfspaces are an example for such desirable classifiers.

Utility/Feasibility Trade-off

Often, learning classes that are desirable for prediction requires large amounts of labeled training data.

On the other hand, there are methods that need less labeled input data but output obscure predictors.

Utilizing unlabeled data to help proper learning

As opposed to labeled training data, unlabeled data is often abundantly available.

As our first demonstration of the potential use of unlabeled samples, we show how such samples be utilized to overcome the utility/feasibility trade-off.

Given a class of desirable classifiers, we will show how unlabeled samples can help find a good classifier from that class.

Our Formal Model: Proper SSL learning

Domain: $\mathcal{X} = [0,1]^d$
Label set: $\{0,1\}$
Target Distribution: P over $\mathcal{X} \times \{0,1\}$, and let D_P denote its induced (marginal) distribution over $\mathcal{X}$

Problem:

Given: A class of desirable classifiers: $H \subseteq 2^{\mathcal{X}}$
A labeled sample S *i.i.d.* from P
A (large) unlabeled sample T *i.i.d.* from D_P

Goal: Output a classifier h from the class H with as small as possible error w.r.t. P.

Our algorithmic paradigm

Step 1 Use the labeled sample to learn a classifier from a *surrogate class* H' (which is easy to learn) that has small prediction error.

Step 2 Apply that learned classifier to label the points of the unlabeled input sample, and feed that now-labeled sample to a fully supervised H-learner (where H is the target class of useful predictors).

Previous Work

This idea of using unlabeled data to transfer learning from one class to another has been investigated before, e.g.

Liang et al., 2008 Used this paradigm to learn fast independent logistic regression classifiers using expressive conditional random fields as a surrogate class.

Bucila et al., 2006 Used a similar idea for an NLP task to get compact neural network predictors.

Overview

We analyze two instantiation of our algorithmic framework:

1. Assuming the data is *realizable* by larger, surrogate class of finite VC-dimension
   - We provide upper bounds and lower bounds on the labeled sample complexity
   - Analyze under which conditions our paradigm yields a saving on labeled examples
   - Show a scenario where unlabeled data *provably* saves labeled data
2. Assuming the data satisfies a relaxed *cluster assumption*
   - We introduce a new, quantitative measure of clusterability
   - We analyze our paradigm using a nearest neighbor algorithm in the first step and provide upper bounds on the sample complexity

Scenario 1: SSL with an approximation class

We assume that the data is realizable by a class H' of finite VC dimension.

The algorithm $A_{(H,H')}$:
Given a labeled sample S and an unlabeled sample T

- First learns a classifier $h' \in H'$ with $\mathrm{Err}_P(h') \leq \epsilon/3$ using S
- Labels T with h'
- Feeds the now-labeled set T to an agnostic learner for H

(Agnostic learners are robust with respect to slight changes in the labeling of the input distribution).

Upper bound

Theorem

For every $\epsilon, \delta \in (0,1)$ *and every target distribution* P*, if* $\mathrm{opt}_{H'}(P) = 0$ *then, given access to a labeled sample* S *of size*

$$|S| \geq \frac{12}{\epsilon}\left(\mathrm{VCdim}(H')\log(36/\epsilon) + \log(4/\delta)\right)$$

and an unlabeled sample T *of size*

$$|T| \geq \frac{576}{\epsilon^2}\left(2\mathrm{VCdim}(H)\log(36/\epsilon) + \log(4/\delta)\right),$$

with probability at least $(1-\delta)$*,*

$$\mathrm{Err}_P(A_{(H,H')}(S,T)) \leq \mathrm{opt}_H(P) + \epsilon.$$

Savings in labeled data

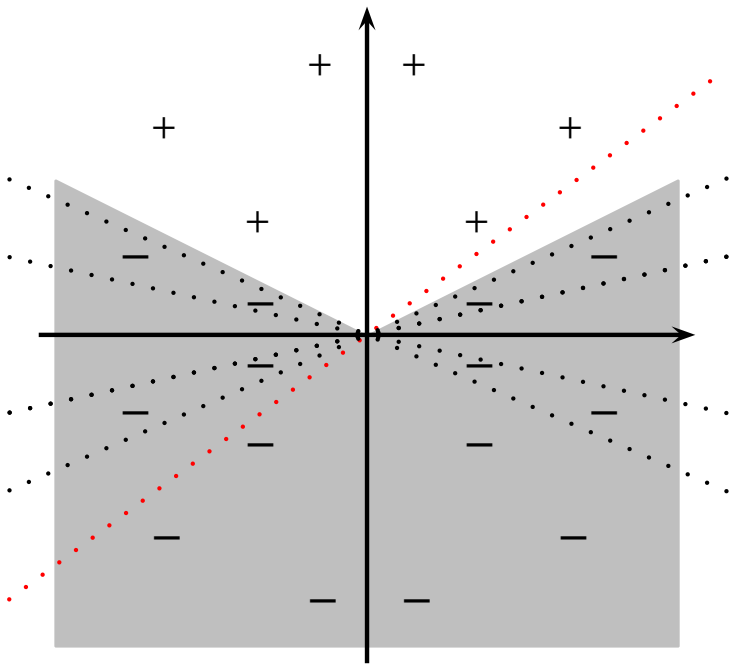
The paradigm saves labels if the sample complexity of learning the surrogate class H' is lower than that of learning the target class H.

There are several possible reasons for a faster convergence rate of H' (or can be learnt faster in terms of computation time):

- Due to lower approximation error (our example)
- Due to lower VC-dimension
- Due to some other exploitable structure of the data

An example scenario where unlabeled data provably helps

Let H' be the class of unions of pairs of halfspaces and H be the class of halfspaces in $\mathbb{R}^d$.



We need to estimate the marginal distribution to decide which halfspace predicts best.

A lower bound on the sample complexity without unlabeled data

$A_{(H',H)}$ uses $O(1/\epsilon)$ labeled examples but any algorithm that has access only to labeled data needs a sample of size $\Omega(1/\epsilon^2)$.

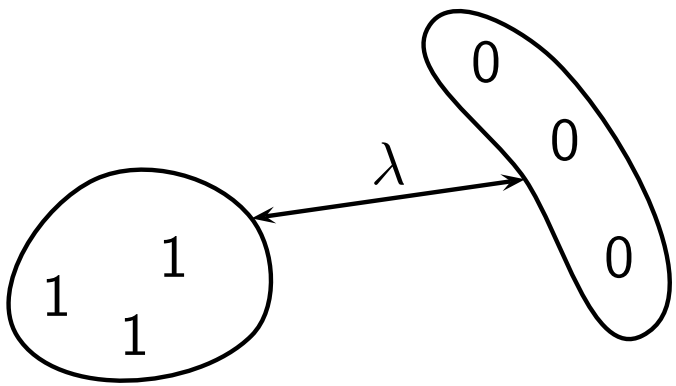
Theorem

The sample complexity of (ϵ, δ)-agnostically learning half-spaces in $\mathbb{R}^d$ over the set of distributions that are realizable by H', is lower bounded below by

$$\frac{1-(1.5\epsilon)^2}{2(1.5\epsilon)^2} \ln\left(\frac{1}{8\delta(1-2\delta)}\right).$$

Scenario 2: SSL with the cluster assumption

If the data is clusterable, the labeling function satisfies a Lipschitz-condition:



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

A new formalization of the cluster-assumption

Definition

For a monotonically increasing $\phi : \mathbb{R}^+ \to [0, 1]$, we say that a function $l : X \to \{0, 1\}$ is ϕ-Lipschitz w.r.t. a probability distribution P over X, if for all $\lambda > 0$,

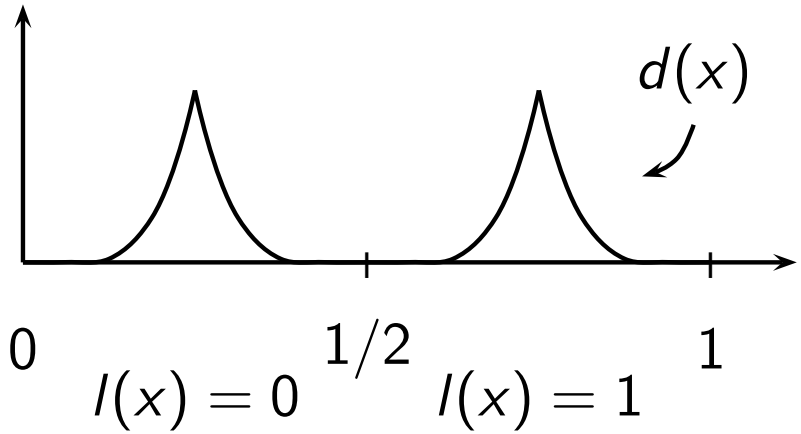
$$\Pr_{x \sim D} \left[\exists y \middle| \; l(x) \neq l(y) \wedge ||x - y|| \leq \lambda \right] \leq \phi(\lambda)$$

.

Intuitively, this implies low density around the label boundaries.

Example–Smoothly clustered data

Domain: $[0, 1]$
Density:
$d(x) = ce^{-1/x}$ for $0 \leq x \leq 1/4$
$d(x) = ce^{-1/|1/2-x|}$ for $1/4 \leq x \leq 3/4$
$d(x) = ce^{-1/|1-x|}$ for $3/4 \leq x \leq 1$



Not λ-Lipschitz for any constant λ.

But satisfies the new measure of clusterability:

$$\Pr_{x \sim D}\left[\exists y \middle| f(x) \neq f(y) \wedge ||x - y|| \leq \lambda\right]$$
$$\leq \int_{1/2-\lambda}^{1/2+\lambda} ce^{-1/|1/2-x|} dx \leq e^{-1/\lambda}$$

SSL with nearest neighbors

A SSL algorithm that exploits our clusterability notion to save labeled examples:

The algorithm A_{NN}
Given a labeled sample S and an unlabeled sample T

1. Labels each example in T with the label of its nearest neighbor in S.
2. Feeds the now-labeled set T to an agnostic learner for H.

Upper bound on the sample complexity for clusterable data

Theorem

Let X be the unit cube of $\mathbb{R}^d$ and let H be any hypothesis class over X with a sample complexity of $m(\epsilon, \delta)$. Given $\epsilon, \delta \in (0, 1)$ and $\lambda > 0$, for every target distribution P over $X \times \{0, 1\}$, whose corresponding labeling function is λ-Lipschitz, with probability at least $1 - \delta$ over the choice of

$$4\left(\sqrt{d}/\lambda\right)^d \frac{6}{\epsilon\delta \mathrm{e}}$$

i.i.d. labeled samples and $m(\epsilon/3, \delta/2)$ i.i.d. unlabeled samples we have that $\mathrm{Err}_P(A_{NN}(S, T)) \leq \mathrm{opt}_H(P) + \epsilon$.

Upper bound on the sample complexity for data satisfying the Probabilistic Lipschitzness

Theorem

Let X be the unit cube of $\mathbb{R}^d$ and let H be any hypothesis class over X with a sample complexity of $m(\epsilon, \delta)$. Given $\epsilon, \delta \in (0,1)$ and $\lambda > 0$, for every target distribution P over $X \times \{0,1\}$, whose corresponding labeling function is ϕ-Lipschitz, where $\phi(a) = e^{-\frac{1}{a}}$, with probability at least $1-\delta$ over the choice of

$$\frac{\sqrt{d}^{5d}}{\epsilon\delta}\left(3\ln(3d^{3/2}(\epsilon\delta)^{-1/d}\right)^d$$

i.i.d. labeled samples and $m(\epsilon/3, \delta/2)$ i.i.d. unlabeled samples we have that $\mathrm{Err}_P(A_{NN}(S,T)) \leq \mathrm{opt}_H(P) + \epsilon$.

Experiments–Setup

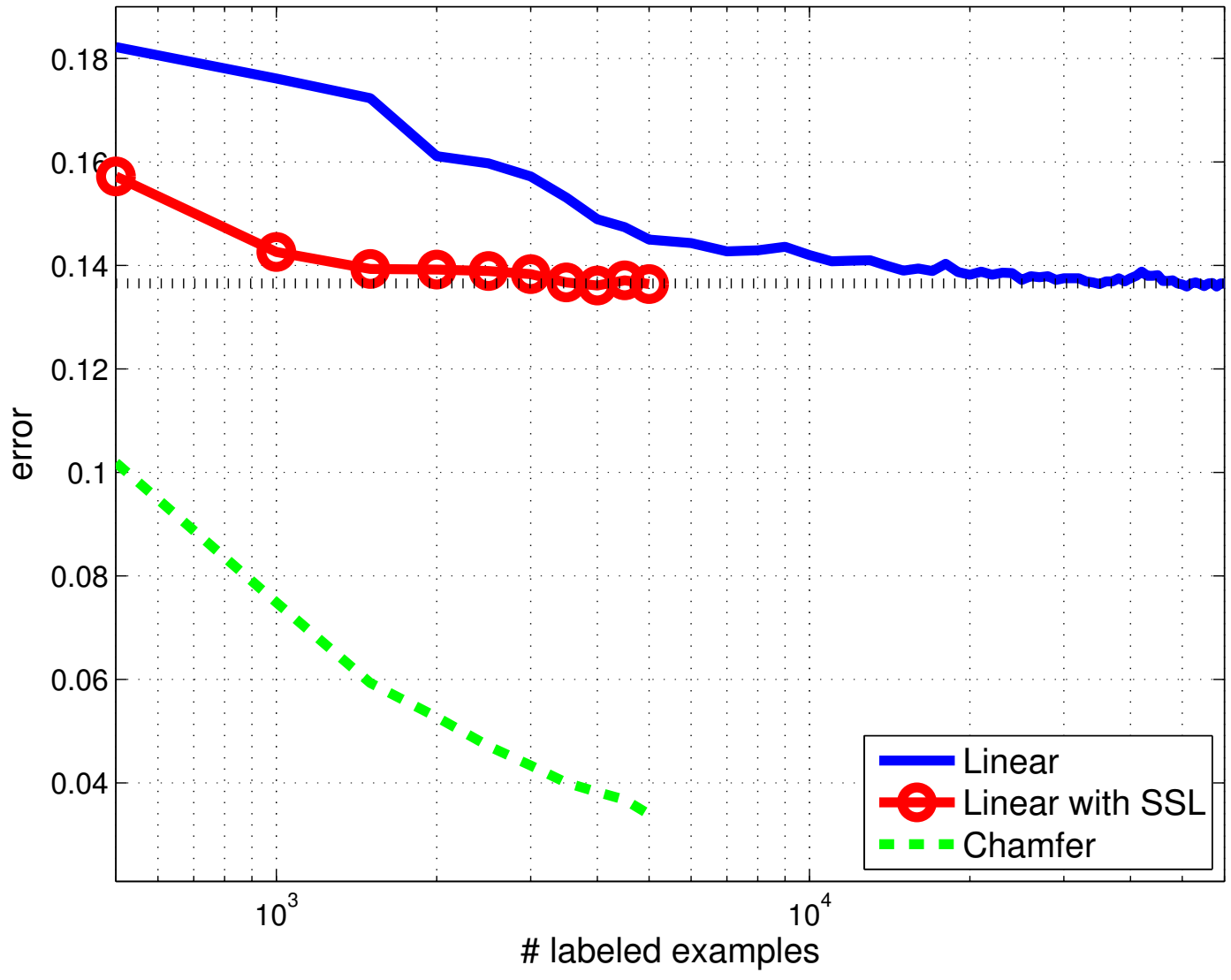
Learning a linear classifier for the MNIST digit recognition dataset (70000 images of digits 0-9)

- Binary classification (0-4) and (5-9)
- Division: 60000 training examples 10000 test examples
- Number of labeled examples: 500-5000 (remaining unlabeled)
- Target class: Linear classifiers in $\mathbb{R}^{28^2}$
- Surrogate class: Kernel based linear classifier with kernel function

$$e^{-d(x,x')/\sigma}$$

where $d(x, x')$ is the *Chamfer distance* between images

Experiments-Results

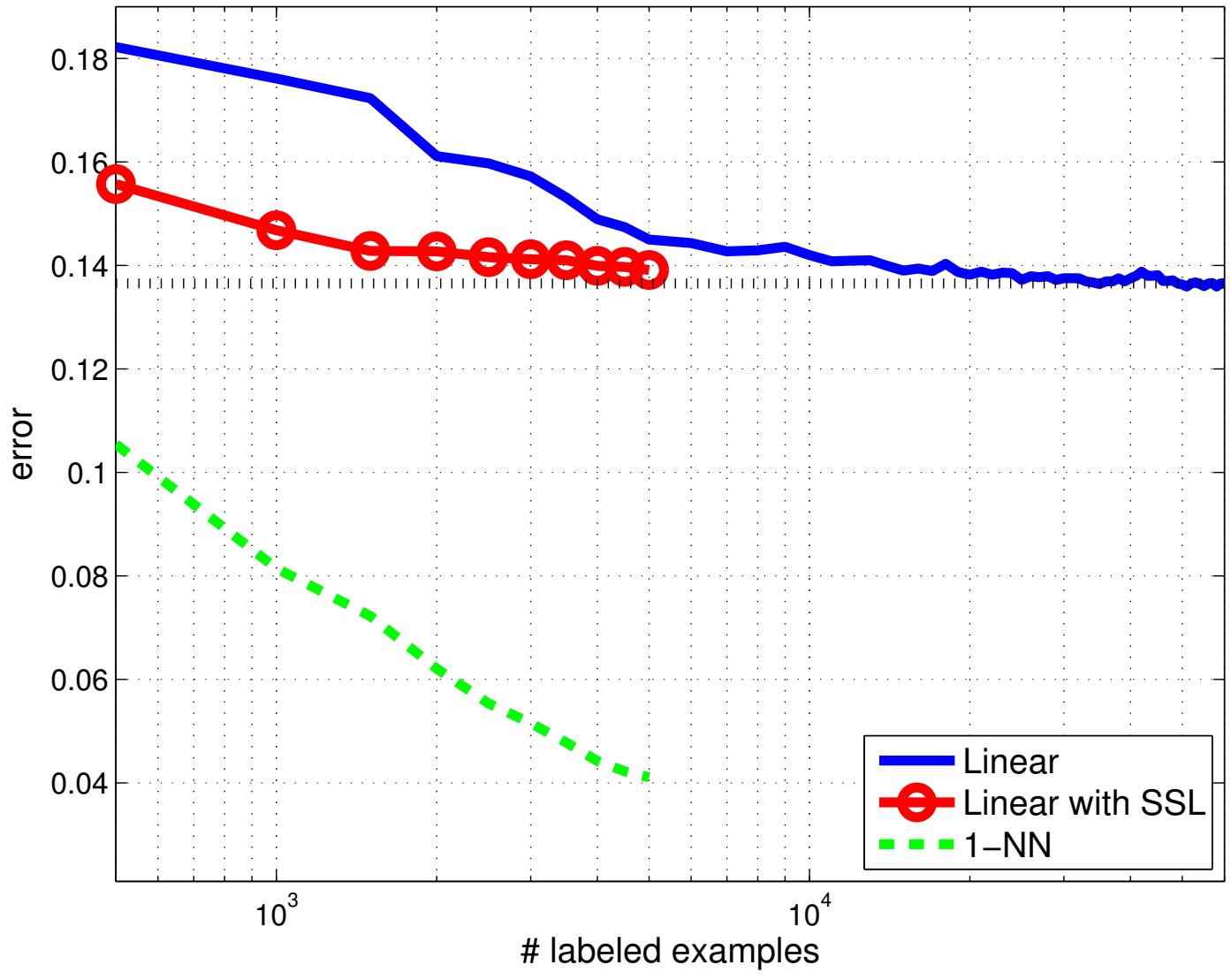


Experiments with NN

Learning a linear classifier for the MNIST digit recognition dataset (70000 images of digits 1-9)

- Binary classification (0-4) and (5-9)
- Division: 60000 training examples 10000 test examples
- Number of labeled examples: 500-5000 (remaining unlabeled)
- Target class: Linear classifiers in $\mathbb{R}^{28^2}$
- Surrogate class: Nearest Neighbors with respect to L_2 metric

Experiments–Results with NN



How many unlabeled examples are needed?

Note that the second stage of our learning paradigm can be formulated independently as a Known Label Classification Learning (we call it KLCL) problem:

Given *a class H of classifiers over some domain X, and a labeling function $f : X \to \{0, 1\}$,*

On **input** *$S \subseteq X$ sampled i.i.d. by some unknown probability distribution, D over X,*

find *some $h \in H$ that minimizes the prediction error w.r.t. (D, f). Namely minimizes $E_D^f(h) = D(\{x \in X : h(x) \neq f(x)\})$.*

Our results (BD and Ben-David, ALT 2011)

We found a simple combinatorial parameter of classes H that characterizes the sample complexity of learning H in this KLCL model.

The size of the sample needed for KLCL of a class H, can either be zero, or $\Theta(1/\epsilon)$, or $\Omega(1/\epsilon^2)$

Where ϵ is an upper bound on the *regret* of the chosen classifier h. Namely, $E_D^f(h) \leq \inf\{E_D^f(h') : h' \in H\} + \epsilon$.

This trichotomy is fully determined by the combinatorial characterization of H.

Use of unlabeled sample for Domain Adaptation

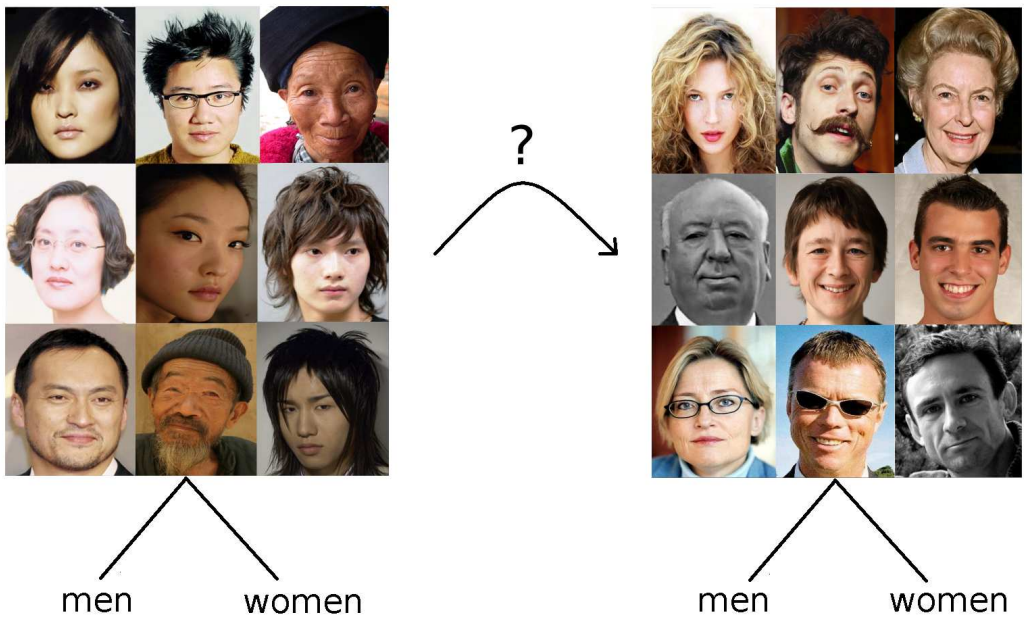
Most of the statistical learning guarantees are based on assuming that *the learning environment is unchanged throughout the learning process.*

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:



- Driving lessons – train on one car, test on another.
- 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.

Domain Adaptation

Domain Adaptation (DA) aims to use training data from some source distribution to construct a good predictor for a different target distribution.

DA works in praxis:

1. McClosky et al. 2010 (NLP - parsing)
2. Blitzer et al. 2006 (NLP - Part of speech tagging)
   - Train on one type of documents (e.g. medical documents)
   - Test on a different type of documents (e.g. legal documents)

But only little theoretical understanding.

Three aspects determining a DA framework

1. The assumptions on the relationship between the source (training) and target (test) data-generating distributions.

2. The type of input data available to the learner.

3. The prior knowledge about the task that the learner has.

Our Model (the input available to the learner)

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

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

Some source-target relatedness assumption

1. *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).
2. *Pointwise weight ratio:* We assume that, for some positive constant, C, for every (measurable) domain subset A, its probability according to the source marginal distribution is at least C times its probability according to the target marginal distribution. (Since otherwise, there may be domain regions that are significant from the target point of view but the labeled training sample is not likely to intersect them)

Conservative vs. Adaptive algorithms

We say that a DA algorithm is *conservative* if it uses only the source-generated labeled sample (and ignores the target-generated unlabeled sample).
We say that a DA algorithm is *adaptive* if its outputs depends on the input target-generated unlabeled sample.

Previous work on conservative algorithms

- BD, Blitzer, Kramer and Pereira (2006) Upper bounds the target-error of ERM on the source in terms of source-target relatedness parameters.
- BD, Shalev-Shwartz, and Urner (2011) Show that under a Lipschitzness and a weight ratio assumption, a Nearest Neighbor classifier from a source sample has low target error.

Are there better adaptive DA algorithms?

- Can we do better than learning on the source domain and applying the SAME hypothesis to the target domain?
- We should! But how?
- Afshani-Mohari-Mansour (2009) propose:

Use the target unlabeled sample to reweigh the training sample. Choose h that minimizes the training error w.r.t. this reweighed training sample.

- Ben-David et al. (2010): It may fail badly!
- Can the unlabeled target sample be unitlized to provably improve performance?

The prior knowledge about the task that the leaner 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, there is no benefit from the use of unlabeled target-generated samples.

Finding the best hypothesis w.r.t. the source training data is as good as any algorithm can do.

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

The algorithmic idea

Given a labeled source-generated sample S and an unlabeled target-generated sample T,
find $h \in H_T$ that has zero training error over $\{(x, \ell) \in S : \ x \in T\}$.

In other words, we use the unlabeled sample T to reweigh the training sample S, by throwing away every member of S that is not hit by a sample point from T.

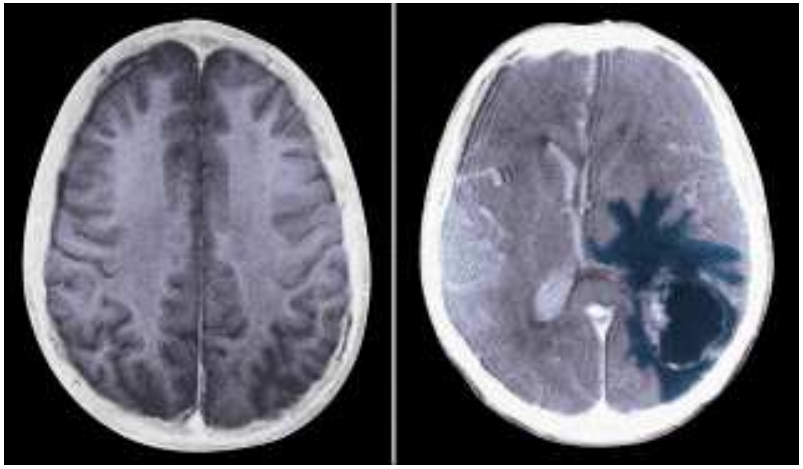
(Note that assuming that H_T has zero approximation error w.r.t. the target distribution, does not rule out the possibility that no member of H_T has zero error over a training sample S that is drawn by the source distribution. However, once S is "cleaned up" using the target sample T, this may no longer happen).

Results

- Having access to sufficiently large target-generated samples, $O\left(\frac{VCdim(H_T)}{\epsilon}\right)$ source-generated training examples suffice to obtain a hypothesis with error $\leq \epsilon$ w.r.t. the target distribution.
- Without any unlabeled target-generated samples, the training sample size required is $\Omega(\sqrt{|X|})$ (where X is the domain set of instances to be labeled).

Learning from Weak Teachers

Consider learning to diagnose brain tumors from CT scans of the skull.



High-accuracy supervision is scarce and expensive. However, supervision by medical students, which is prone to errors, maybe obtained cheaply.

Can labels generated by such *weak teachers* be utilized to save queries to a precise, but expensive, supervisor?

Modeling weak teachers

Goal: Learn a good label predictor from random examples that are labeled by two kinds of teachers:

- *strong teacher* labels correctly,
- *weak teacher* may make mistakes, but can handle "clear cut" cases.

The learning program can, for any given image, choose which teacher to query about its label.

Question 1: How should we model supervision provided by novices?
Question 2: Can access to a weak teacher's labels save queries to a strong teacher?

Overview

We can replace expert knowledge under certain conditions:

- Property 1 Weak teachers label clear cut cases correctly (with high probability).
- Property 2 Weak teachers give both labels (but not necessarily the correct ones) in borderline areas.
- Property 3 There are not too many borderline areas.

Our requirements from weak teachers

Rather than *defining* how should weak teachers predict, we allow any prediction rule that satisfies some requirements.

Our requirements are formulated in terms of the expected label of 'neighboring instances' (weighted by their similarity to the instance we wish to label).

1. for instances having that expected label close to either 0 or 1, the weak teacher should provide that value (with high probability).
2. The weak teacher should "hesitate" when labeling instances for which that expected label over the similarity neighborhood is close to $1/2$ (that is, the similarity neighborhood is label-heterogeneous).

Our results - utilizing weak-teacher's labels

We prove that whenever the underlying data-generating distribution satisfies some mildness conditions, labels generated by any rule satisfying the above requirements can be utilized for learning a low-error prediction rule while saving queries for correct labels.

The algorithmic paradigm

1. Obtain two random samples of domain points, S and T (sampled i.i.d. according to the marginal distribution) and query the weak teacher for the labels of the points in S.
2. Use these labels to estimate our confidence in the correctness of the labels that would be assigned by the weak teacher to each point of T.
3. Query the strong teacher about the labels of points in T for which that confidence is low.
4. Label the high confidence points of T using the weak teacher's labels on the sample S.
5. Pass T with the labels obtained (by this procedure) to an agnostic learner.

Result

Assume there is a learning algorithm that achieves an error rate of at most ϵ using a correctly labeled sample of size $m(\epsilon)$, then, using this algorithm as a subroutine, our algorithm makes only $\psi(\eta)m(\epsilon)$ queries to the strong teacher and outputs a classifier whose error is at most $\epsilon + 2\varphi(\eta)$.

	Standard Learner	Our Algorithm
Error	ϵ	$\epsilon + 2\varphi(\eta)$
Number of strong queries	$m(\epsilon)$	$\psi(\eta)m(\epsilon)$
Number of weak queries	0	many

If the distribution is very local conservative and very nice (*i.e.* $\varphi(\eta)$ and $\psi(\eta)$ are small), then our algorithm saves queries to the strong teacher while only increasing the error insignificantly.

Conclusions

In many machine learning applications, unlabeled, or weakly labeled examples are way more readily available then correctly labeled examples.

We look for ways to utilize such cheap resource to help classification prediction.

In this talk I described three scenarios in which this *provably* works:

1. **"Proper" Semi Supervised Learning**
2. **Domain Adaptation**
3. **Learning for Weak Teachers**

For each of these tasks we proposed an algorithmic paradigm that utilizes the unlabeled (or weakly labeled) samples.

Open questions and research challenges

Don't be fooled: This is only a theoretical research.

The real remaining challenge is to find ways to make these paradigms work in practice

I hope that in the future I will hear from you if and how these ideas can be implemented in the real world.