Boosting for Domain Adaptation

A. Habrard, J.P. Peyrache, M. Sebban

Laboratoire Hubert Curien, UMR CNRS 5516
University of Jean Monnet Saint-Étienne (France)

September, 2012
Supervised learning problem

**Notations**

- Let $S$ be a set of $m$ training examples $\{z_i = (x_i, y_i)\}_{i=1}^m$ i.i.d. from an unknown distribution $D_Z$ over a space $Z = \mathcal{X} \times \mathcal{Y}$.
- The $x_i$ values ($x_i \in \mathcal{X}$) are typically feature vectors.
- In classification, the $y$ values ($y \in \mathcal{Y}$) are drawn from a discrete set of classes (typically $\mathcal{Y} = \{-1, +1\}$ in binary classification).
- We assume that there exists an unknown function $f$ such that $y = f(x)$, $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$.

**Definition**

A supervised learning algorithm $L$ automatically outputs from $S$ a classifier (or a hypothesis) $h \in \mathcal{H}$ as close as possible to the target function $f$. 
True risk and empirical risk

The goal is to find a hypothesis $h$ that achieves the smallest true risk $\mathcal{R}^l(h)$.

**Definition (True Risk)**

The true risk $\mathcal{R}^l(h)$ (also called generalization error) of a hypothesis $h$ w.r.t. a loss function $l : \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}^+$ corresponds to the expected loss suffered by $h$ over $\mathcal{D}_\mathcal{Z}$.

$$\mathcal{R}^l(h) = \mathbb{E}_{(x,y) \sim \mathcal{D}_\mathcal{Z}} [l(h, z)].$$

**Nota**

The most natural loss function for binary classification is the 0/1 loss:

$$l_{0/1}(h, z) = 1 \text{ if } yh(x) < 0 \text{ and } 0 \text{ otherwise.}$$

$\mathcal{R}^{l_{0/1}}(h)$ then corresponds to the classification error.
Introduction

True risk and empirical risk (ctd)

Nota

Unfortunately, $\mathcal{R}^l(h)$ cannot be computed because $\mathcal{D}_Z$ is unknown. We can only measure it on the training set $S \leftrightarrow$ empirical risk $\hat{\mathcal{R}}^l(h)$.

Definition (Empirical Risk)

Let $S = \{z_i = (x_i, y_i)\}_{i=1}^m$ be a training sample. The empirical risk $\hat{\mathcal{R}}^l(h)$ (also called empirical error) of a hypothesis $h \in \mathcal{H}$ with respect to a loss function $l$ corresponds to the expected loss suffered by $h$ on the instances in $S$.

$$
\hat{\mathcal{R}}^l(h) = \frac{1}{m} \sum_{i=1}^m [l(h, z_i)]
$$
Due to the nonconvexity of the 0/1 loss, minimizing $\hat{R}^{0/1}(h)$ is known to be NP-hard even for simple classes of hypotheses (Ben-David et al., 2003).

For this reason, surrogates convex loss functions are often used, e.g.:

- the hinge loss (used in SVM):
  \[ l_{\text{hinge}}(h, z) = [1 - yh(x)]_+ = \max(0, 1 - yh(x)), \]

- the logistic loss (used in logistic regression):
  \[ l_{\log_2}(h, z) = \log_2(1 + e^{-yh(x)}), \]

- the exponential loss (used in Adaboost):
  \[ l_{\text{exp}}(h, z) = e^{-yh(x)}. \]
Choosing an appropriate loss function is not an easy task and heavily depends on the problem at hand.

Minimizing $\hat{R}^l(h)$ is a very common strategy but requires some care - $\hat{R}^l(h)$ is usually too optimistic (one can learn by heart!).
Occam’s razor

Most used inductive principle in ML: Occam’s razor

“Choose the simplest explanation consistent with past data’’.

“no sunt multiplicanda entia praeter necessitatem” (William of Ockham)  
(Entities are not to be multiplied beyond necessity)

W. Ockham: Born around 1285, he was an English philosopher and monk.

Occam’s razor through the ages...
Updated definition

Learning can be viewed as a problem of function estimation, whose goal is to determine the simplest consistent (w.r.t. S) hypothesis $h \in \mathcal{H}$.

Assumption in supervised learning

“What you see (in learning) is what you get (in deployment)”.

In other words: both training and test data are drawn according to $\mathcal{D}_Z$. 
Introduction

Domain Adaptation

Alert

Life is not so simple...
The previous assumption does not hold in many real applications!

This requires Domain Adaptation!

New assumption

“What you SAW (in learning) is NOT what you get (in deployment)”.

≃ training and test data are drawn according to different distributions.
Domain Adaptation (DA)

Definition

A standard DA problem can be defined as a situation where the learner receives labeled data drawn from a source domain and very few or no labeled points from the target distribution.

Conjecture

Boosting, which is known to optimize any learning algorithm and which maximizes the margins, is a good candidate to adapt well when the training set is made of unlabeled target data.

The aim of the rest of this talk is to prove this proposition!
Theory of 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

**Input:** A learning sample $S$, a number of iterations $N$, a weak learner $L$

**Output:** A global hypothesis $H_N$

for $i = 1$ to $m$ do
  $D_1(x_i) = 1/m$
end for

for $t = 1$ to $N$ do
  $h_t = L(S, D_t)$
  $\hat{\epsilon}_t = \sum_{x_i \text{ s.t. } y_i \neq h_t(x_i)} D_t(x_i)$
  $\alpha_t = \frac{1}{2} \ln \frac{1 - \hat{\epsilon}_t}{\hat{\epsilon}_t}$
  for $i = 1$ to $m$ do
    $D_{t+1}(x_i) = D_t(x_i) \exp \left( -\alpha_t y_i h_t(x_i) \right) / Z_t$
    /* $Z_t$ is a normalization coefficient*/
  end for
end for

$f(x) = \sum_{t=1}^{N} \alpha_t h_t(x)$

$H_N(x) = \text{sign} \left( f(x) \right)$
Theory of Boosting

Theoretical result on the empirical error

**Theorem**

**Upper bound on the empirical error of the final classifier** $H_T$

\[
\hat{\epsilon}_{H_N} \leq \prod_{t=1}^{N} Z_t \leq \exp\left(-2 \sum_{t=1}^{N} \gamma_t^2\right)
\]

where $\hat{\epsilon}_t = \frac{1}{2} - \gamma_t$ (weak hypothesis).

$\hat{\epsilon}_{H_N}$ is optimized with $\alpha_t = \frac{1}{2} \ln \frac{1 - \hat{\epsilon}_t}{\hat{\epsilon}_t}$.

**Meaning**

This theorem means that the empirical error **exponentially decreases towards 0** with the number $N$ of iterations.
Explanation in terms of margins of the training examples

After \( n \) iterations

\[
\gamma
\]

After \( n' > n \) iterations

Theorem

\[
\forall \theta > 0, \text{ with probability } 1 - \delta, \text{ any classifier ensemble } H_N \text{ satisfies:}
\]

\[
\epsilon_{H_N} \leq \mathbb{E}_{x \in S}[\text{margin}(x) \leq \theta] + O \left( \sqrt{\frac{d_h \log^2(m/d_h)}{m} \frac{\theta^2}{\theta^2} + \log(1/\delta)} \right),
\]

where \( \mathbb{E}_{x \in S}[\text{margin}(x) \leq \theta] \) exponentially decreases towards 0 with \( N \).
Conclusion

- **ADABOOST works in practice**
- ... is theoretically well-founded!
- ... is by nature adaptive,
- ... maximizes the margins.

Claim

Boosting seems to be a good candidate to deal with *Domain Adaptation*. 
DABoost
**Scenario**

We focus on the scenario where the training set is made of **labeled source** data (black circles and triangles) and **unlabeled target** instances (white squares).

**Graphical illustration of the strategy**
Weak DA hypothesis

A classifier \( h_n \) learned at iteration \( n \) is a weak DA hypothesis for \( T \) if:

1. \( h_n \) is a weak classifier for \( S \).
2. \( \exists \tau_n^T \in ]0; \frac{1}{2} \] s.t. \( \hat{L}_n = \mathbb{E}_{x_i \sim D_n^T} [|h_n(x_i)| \leq \gamma] = \frac{W_T^-}{W_T} = \frac{1}{2} - \tau_n^T \),

where \( D_n^T(x_i) = \frac{D_n(x_i)}{W_T} \) if \( x_i \in T \), 0 otherwise.

- \( W_{T^+} \) (resp. \( W_{T^-} \)) is the sum of the weights of the target examples outside the margin band \( T^+ \) (resp. inside \( T^- \)). \( W_T = W_{T^-} + W_{T^+} \).

- \( W_{S^+} \) (resp. \( W_{S^-} \)) is the sum of the weights of the source data correctly (resp. incorrectly) classified by the hypothesis \( h_n \).
DABoost

**Input:** two sets $S$ and $T$, a number of iterations $N$, a margin $\gamma \leq 1$, $m = |T| + |S|$.

**Output:** two source and target classifiers $H^S_N$ and $H^T_N$.

Initialization: $\forall x_i \in S \cup T$, $D_1(x_i) = \frac{1}{m}$.

for $n = 1$ to $N$ do

Learn $h_n$ satisfying the weak DA assumption.

$$\alpha_n = \frac{1}{2} \ln \frac{W_{S+}}{W_{S-}} \quad \text{and} \quad \beta_n = \frac{1}{2\gamma} \ln \frac{W_{T+}}{W_{T-}}$$

$\forall x_i \in S$, $D_{n+1}(x_i) = D_n(x_i) \cdot \frac{e^{-\alpha_n \text{sign}(h_n(x_i)) \cdot y_i}}{Z_n}$.

$\forall x_i \in T$, $D_{n+1}(x_i) = D_n(x_i) \cdot \frac{e^{-\beta_n h_n(x_i) \cdot y_i^n}}{Z_n}$, where $y_i^n = \text{sign}(h_n(x_i))$ if $|h_n(x_i)| > \gamma$, $y_i^n = -\text{sign}(h_n(x_i))$ otherwise, and $Z_n$ is a normalization coefficient.

end for

$$f^S_N(x) = \sum_{n=1}^N \alpha_n \text{sign}(h_n(x))$$, and $$f^T_N(x) = \sum_{n=1}^N \beta_n \text{sign}(h_n(x))$$.

Final source and target classifiers:

$H^S_N(x) = \text{sign}(f^S_N(x))$ and $H^T_N(x) = \text{sign}(f^T_N(x))$. 
Illustration of DABoost on an artificial example

We consider two source moons in a 2-dimensional space. The target domain is obtained by rotating anticlockwise the source domain according to rotation angles from 20 to 50 degrees.

(c) On the source with $f^S_N(x)$

(d) On the target with $f^T_N(x)$

$$f^S_N(x) = \sum_{n=1}^{N} \alpha_n \text{sign}(h_n(x)) \text{ and } f^T_N(x) = \sum_{n=1}^{N} \beta_n \text{sign}(h_n(x)).$$

**Question**

Why does it work so well? → Need to theoretically analyze DABoost.
**Theoretical Analysis of DAB**

**Theorem (Upper bound on the margin loss \( \hat{L}_{H_N^T} \))**

Let \( \hat{L}_{H_N^T} \) be the proportion of target examples of \( T \) with a margin smaller than \( \gamma \) after \( N \) iterations of DABOOST:

\[
\hat{L}_{H_N^T} = \mathbb{E}_{x_i \sim T} [y_i f^N_T(x_i) < 0] \leq \frac{m}{|T|} \Pi_n Z_n,
\]

where \( y_i = (y^1_i, \ldots, y^n_i, \ldots, y^N_i) \) and \( f^N_T(x_i) = (\beta_1 h_1(x_i), \ldots, \beta_n h_n(x_i), \ldots, \beta_N h_N(x_i)) \).

**Theorem (Optimal confidence coefficients)**

The minimization of \( Z_n \) is ensured by assigning the optimal coefficient \( \beta_n = \frac{1}{2\gamma} \ln \frac{W_{T^+}}{W_{T^-}} \) to each weak DA classifier.
Theorem (Convergence of the margin loss)

The following upper bound holds for the empirical loss $\hat{L}_{H_N^T}$ of the final classifier $H_N$:

$$\hat{L}_{H_N^T} \leq \frac{m}{|T|} \exp \left( -2 \sum_n \min(\tau_n^S, \tau_n^T) \right),$$

where $\tau_n^S = \frac{1}{2} - W_S$ and $\tau_n^T = \frac{1}{2} - W_T$. This theorem means that the empirical loss $\hat{L}_{H_N^T}$ decreases exponentially fast towards 0 with $N$. 

A. Habrard, J.P. Peyrache, M. Sebban (LAHC)  
Boosting for Domain Adaptation  
September, 2012  
23 / 29
Illustration on a $20^\circ$ rotation
Theoretical Analysis of DABoost

Theorem (Upper bound on the target generalization error $\epsilon_H^T$)

The final classifier learned after $N$ iterations of DABoost satisfies on the target generalization error $\epsilon_H^T$ the following bound:

$$\epsilon_H^T \leq \hat{L}_S(H_N) + \frac{1}{2}\hat{d}_{\mathcal{H},\gamma}(S, T) + \lambda_* + C_1,$$

where:

- $\hat{L}_S(H_N) = \mathbb{E}_{x_i \sim S}[y_if_N^S(x_i) \leq \gamma]$ is known to decrease with Adaboost.

- $\lambda_*$ is the loss of the ideal joint hypothesis (supposed to be small when adaptation is possible).

- $C_1$ is a constant function in $m = |T| + |S|$ and $\mathcal{H}$.

- $\frac{1}{2}\hat{d}_{\mathcal{H},\gamma}(S, T)$ is a divergence between $S$ and $T$. 
A solution (inspired from Ben David et al. 2010) consists in projecting both source and target data in the one-dimensional space given by $f(x) = \sum_n (\alpha_n + \beta_n) h_n(x)$. Then, $\hat{d}_{\mathcal{H}_\gamma}(S, T)$ is computed in that space by calculating the proportion of margin disagreements between $S$ and $T$ as follows:

$$
\hat{d}_{\mathcal{H}_\gamma}(S, T) = \sup_{h \in \mathcal{H}} [E_{x \sim S}[|f(x)| \leq \gamma] - E_{x \sim T}[|f(x)| \leq \gamma] + [E_{x \sim S}[|f(x)| > \gamma] - E_{x \sim T}[|f(x)| > \gamma]].
$$
We compared DABoost with two widely used DA approaches (DASVM and a reweighting approach for the sample selection bias problem - SVM-W).

<table>
<thead>
<tr>
<th>Angle</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>Average</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>10.3</td>
<td>24</td>
<td>32.2</td>
<td>40</td>
<td>26.6 ± 0.9</td>
<td></td>
</tr>
<tr>
<td>AdaBoost</td>
<td>20.9</td>
<td>32.1</td>
<td>44.3</td>
<td>53.7</td>
<td>37.8 ± 0.4</td>
<td></td>
</tr>
<tr>
<td>DASVM</td>
<td>0.0</td>
<td>21.6</td>
<td>28.4</td>
<td>33.4</td>
<td>20.9 ± 3.2</td>
<td></td>
</tr>
<tr>
<td>SVM-W</td>
<td>6.8</td>
<td>12.9</td>
<td>9.5</td>
<td>26.9</td>
<td>14.0 ± 5.3</td>
<td></td>
</tr>
<tr>
<td>DABoost</td>
<td>2.3</td>
<td>4.4</td>
<td>7.0</td>
<td>11.0</td>
<td>6.2 ± 1.2</td>
<td></td>
</tr>
</tbody>
</table>
Conclusion

- Domain Adaptation arises in a large spectrum of applications, such as in computer vision, speech processing, natural language processing.

- These applications challenge the common learning theories (e.g. the PAC model).

- The new theory of DA opens the door for designing theoretically well-founded DA-algorithms.

- By maximizing the margins, boosting is a good candidate to deal with DA problems.
Ultimate conclusion

“What you SAW (in learning) IS NOT CAN BE THEORETICALLY WELL ADAPTED TO BE what you get (in deployment)”.