Introduction to Statistical Learning Theory

Lecture 9

We return to the binary classification problem.

So far we investigated when is $L_{\mathcal{D}}(A(S))$ close to $L_S(A(S))$, and more impotently to $\min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$ with high probability.

The problem is - how do you build a hypothesis set that has small empirical loss AND generalizes?

Another issue is computational - being able to find a good hypothesis statistically is nice, but in practice you need to find it in a computational efficient manor!

This leads to the idea of boosting. Assume you only have access to a "weak" learner, that can only do a bit better then chance. Can you "boost" its accuracy to get a "strong" leaner?

Notice: In our general framework, even "weak" learning may be impossible

Solution: We will restrict our discussion to data that is labeled by some unknown function $c : \mathcal{X} \to \{\pm 1\}$. i.e. there is an unknown distribution $\mathcal{D}$ on $\mathcal{X}$ and for all $x \sim \mathcal{D}$ we have $y = c(x)$.

Unlike the realizable case, we will not asusme $c \in \mathcal{H}$. We will assume it belongs to some large, known set $\mathcal{C}$ called the concept space.

Definition 1.1 (”strong” learner)

We say algorithm A is a strong learning algorithm for concept class $\mathcal{C}$ if for any distribution $\mathcal{D}$ on $\mathcal{X}$, labeling function $c \in \mathcal{C}$, $0 < \delta < 1$ and $\epsilon > 0$ there exists $\mathcal{M}(\epsilon, \delta)$ such that if the algorithm is given $m > \mathcal{M}(\epsilon, \delta)$ labeled samples from this distribution the algorithm returns a classifier $A(S)$ such that with probability greater or equal to $1 - \delta$ we have $L_{\mathcal{D}}(A(S)) < \epsilon$.

Definition 1.2 (γ-”weak” learner)

We say algorithm A is a γ-weak learning algorithm for concept class $\mathcal{C}$ if for any distribution $\mathcal{D}$ on $\mathcal{X}$, labeling function $c \in \mathcal{C}$ and $0 < \delta < 1$ there exists $\mathcal{M}(\delta)$ such that if the algorithm is given $m > \mathcal{M}(\delta)$ labeled samples from this distribution the algorithm returns a classifier $A(S)$ such that with probability greater or equal to $1 - \delta$ we have $L_{\mathcal{D}}(A(S)) < 1/2 - \gamma$.

The problem: given a weak learner, as a black box, can we "boost" its accuracy and return a strong learner?

We will look at classifiers of the type $H(x) = sign(\sum_i \alpha_i h_i(x))$ where h_i are classifiers returned by the weak learner.

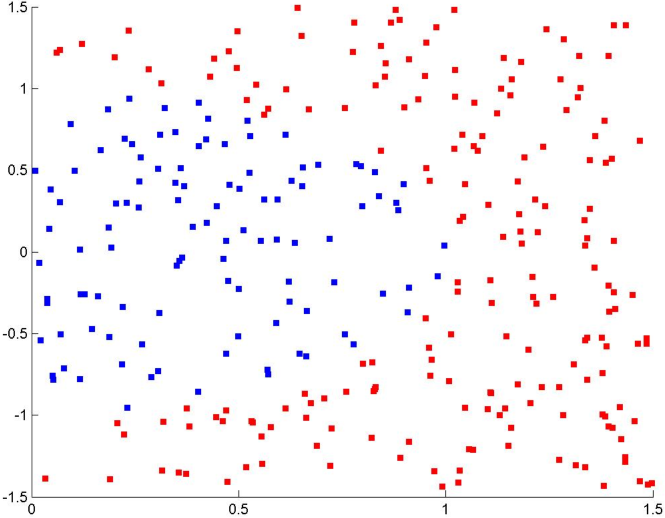
The first practical boosting algorithm is adaBoost (adaptive boosting).

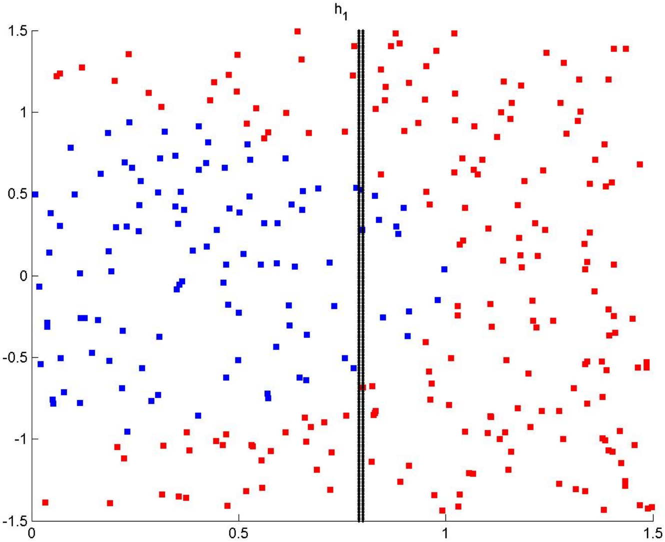
The idea: At each iteration you reweigh the training sample, giving larger weight to points where classified wrongly and give this to the weak learner.

For all sample $S = (x_1, y_1), ..., (x_m, y_m)$ and distribution $\mathbf{D}$ on $(x_1, ..., x_m)$, we define $WL(\mathbf{D}, S)$ the hypothesis returned by the weak learner that tries to minimize $\sum_{i=1}^{m} \mathbf{D}(i)\mathbb{1}[y_i \neq h(x_i)]$.

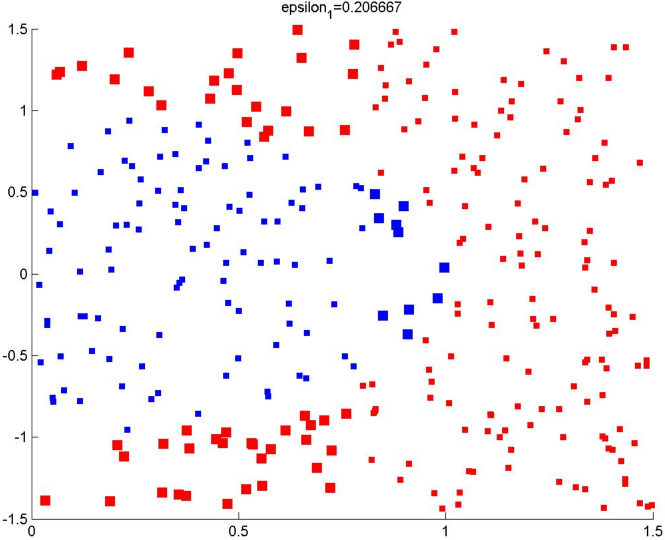
Algorithm `adaBoost`

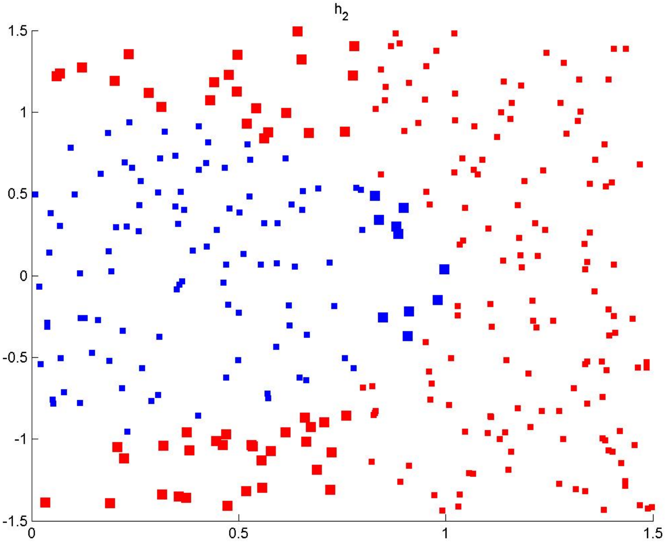
Input: training set $S = (x_1, y_1), ..., (x_m, y_m)$, weak learner WL and number of iteration T.
Initialize: $\mathbf{D}^1 = (\frac{1}{m}, ..., \frac{1}{m})$
for t=1,...,T **do**
  $h_t = WL(\mathbf{D}^t, S)$ % Invoke weak learner
  compute $\epsilon_t = \sum_{i=1}^{m} \mathbf{D}^t(i) \mathbb{1}[y_i \neq h_t(x_i)]$
  compute $\alpha_t = \frac{1}{2}\log(\frac{1}{\epsilon_t} - 1)$
  Update: $\mathbf{D}^{t+1}(i) = \frac{\mathbf{D}^t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$ % Z_t normalizer .
end for
return classifier $H(x) = sign(\sum_i \alpha_i h_i(x))$


1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5

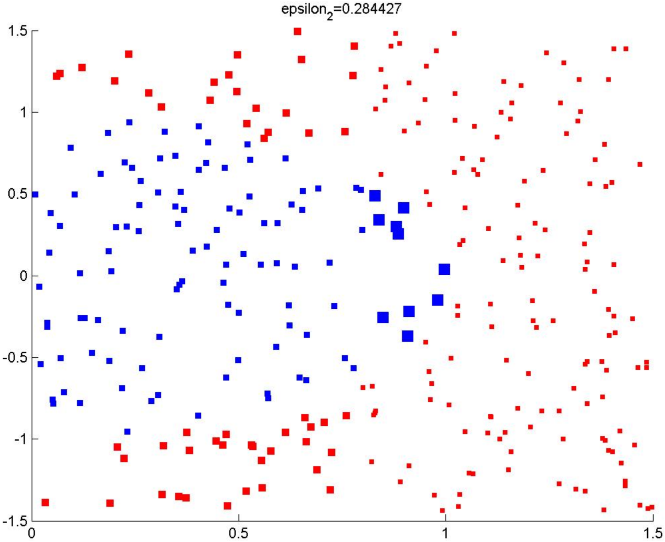

h1
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5

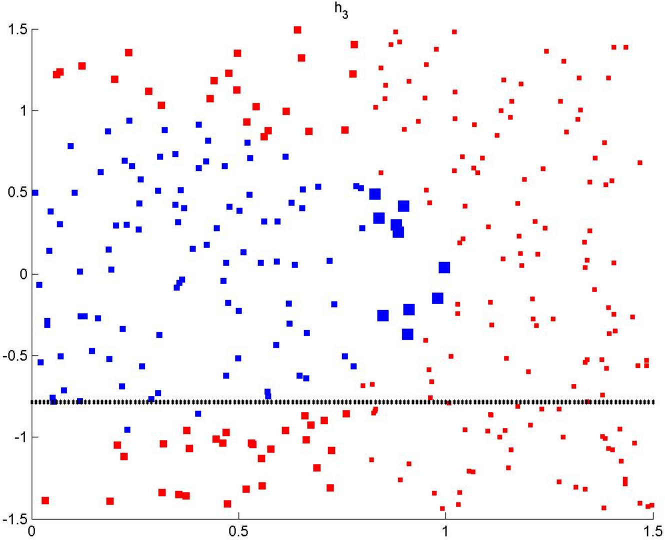
adaBoost algorithm



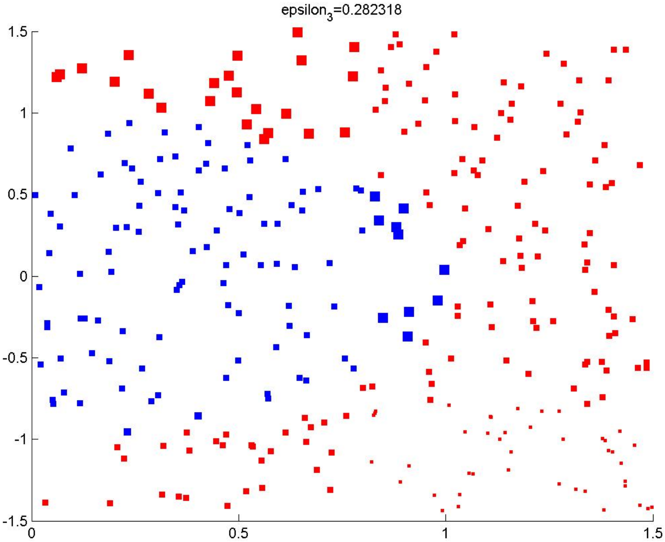

h$_2$
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5

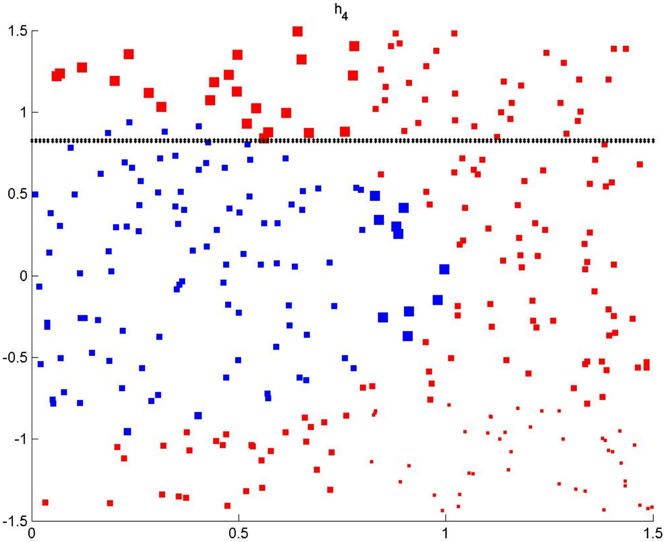
adaBoost algorithm

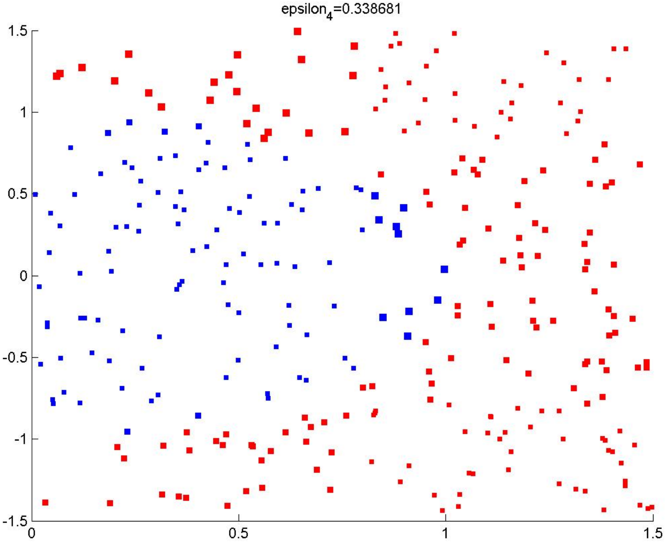



h3
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5

adaBoost algorithm




h4
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5


epsilon$_4$=0.338681
1.5
1
0.5
0
-0.5
-1
-1.5
0
0.5
1
1.5

We now show the loss decays exponentially.

Theorem 2.1

Let ϵ_t be the weak learners error at iteration t and define $\gamma_t = 1/2 - \epsilon_t$. The empirical loss of H is bounded by

$$L_S(H) = Pr_{i\sim\mathbf{D}^1}\left(H(x_i \neq y_i)\right) \leq \prod_{t=1}^{T} \sqrt{1-4\gamma_t^2} \leq \exp\left(-2\sum_{i=1}^{T}\gamma_i^2\right) \quad (1)$$

If we assume a γ-weak learner, we can simplify the bound to $\exp(-2\gamma^2 T)$.

Intuition: H is a (weighted) majority vote. For it to error on x_i, many rounds must be erroneous. This means high (unnormalized) weight, since the weak learner is better then chance the total weight decays and there can be only few elements with large weight.

Proof: Define $F(x) = \sum_{i=1}^{T} \alpha_i h_i(x)$, so $H(x) = sign(F(x))$.

We can rewrite $\mathbf{D}^{T+1}$ using the algorithm recursive formula

$$\begin{aligned}
\mathbf{D}^{T+1}(i) &= \mathbf{D}^{T}(i)\frac{\exp(-y_i\alpha_T h_T(x_i))}{Z_T} \qquad (2)\\
&= \mathbf{D}^{T-1}(i)\frac{\exp(-y_i\alpha_{T-1} h_{T-1}(x_i))}{Z_{T-1}} \cdot \frac{\exp(-y_i\alpha_T h_T(x_i))}{Z_T}\\
&= \mathbf{D}^{1}(i)\frac{\exp\left(-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)\right)}{\prod_{t=1}^{T} Z_t} = \mathbf{D}^{1}(i)\frac{\exp(-y_i F(x))}{\prod_{t=1}^{T} Z_t}
\end{aligned}$$

The next this is to note that $\mathbb{1}[H(x) \neq y] \leq \exp(-yF(x))$.

We can now write the training error as

$$Pr_{i\sim\mathbf{D}^1}\left(H(x_i \neq y_i)\right) = \sum_{i=1}^{m} \mathbf{D}^1(i)\mathbb{1}[H(x_i) \neq y_i] \leq \sum_{i=1}^{m} \mathbf{D}^1(i)\exp(-y_i F(x_i))$$

$$= \sum_{i=1}^{m} \mathbf{D}^{T+1}(i)\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} Z_t \tag{3}$$

Finally we look at Z_t:

$$Z_t = \sum_{i=1}^{m} D_t(i)e^{-\alpha_t y_i h_t(x_i)} = \sum_{y_i = h_t(x_i)} D_t(i)e^{-\alpha_t} + \sum_{y_i \neq h_t(x_i)} D_t(i)e^{\alpha_t}$$

$$= e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t = \sqrt{4\left(\frac{1}{2}-\gamma_t\right)\left(\frac{1}{2}-\gamma_t\right)} = \sqrt{1-4\gamma_t^2} \tag{4}$$

We can show that that α_t minimizes Eq. 4.

We now analyse the VC-dimension of boosting.

Assume the weak learner returns a classifier from a base space B with dimension $VC(B)$.

The boosted classifier "lives" in the following space

$$L(B,T) = \left\{ x \mapsto sign\left(\sum_{i=1}^{T} \alpha_t h_t(x)\right) : \alpha \in \mathbb{R}^T, \forall t, h_t \in B \right\}$$

Theorem 2.2

Assume $VC(B)$ and T are at least 3, then the following holds:

$$VC(L(B,T)) \leq 3T(VC(B)+1) \cdot (\ln(T(VC(B)+1)) + 1)$$

Proof: Denote $d = VC(B)$. Assume we are given inputs $x_1, ..., x_m$. Any classifier in L is a linear hypothesis in the space $(h_1(x), ..., h_T(x))$.

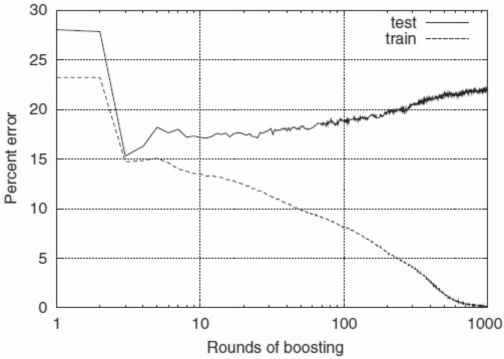
As $d = VC(B)$, from Sauer-Shelach lemma, there are at most $(em/d)^d$ labellings to pick from. This means there are at most $(em/d)^{dT}$ ways to pick T predictors $(h_1(x), ..., h_T(x))$.

Linear predictors in dimension T have VC-dimension T. So for each T predictors we have at most $(em/T)^T$ classifiers, totaling $(em/d)^{dT}(em/T)^T \leq m^{T(d+1)}$. For a set of size m to be shattered we must have $2^m \leq m^{T(d+1)}$ or $m \leq \frac{T(d+1)}{\ln(2)} \ln(m)$.

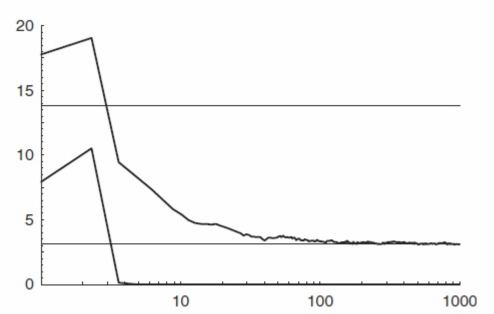
We showed $m \leq \frac{T(d+1)}{\ln(2)} \ln(m)$
Using the lemma (which we will prove shortly) for $a > 0$, $x \leq a\ln(x) \rightarrow x \leq 2a\ln(a)$ we get $m \leq 2\frac{(d+1)T}{\ln(2)} \ln\left(\frac{(d+1)T}{\ln(2)}\right)$ from which we can get our desired bound. □

Proof of the lemma: Assume by contradiction $x \leq a\ln(x)$ and $x > 2a\ln(a)$. This implies $a\ln(x) > 2a\ln(a)$ or $x > a^2$. Define now $x = c \cdot a$, and plug in the second inequality to get $a < e^{c/2}$. Use this in the first inequality to get $c < 2\ln(c)$ which has no solution.

We expect adaBoost to overfit when T grows



Many times this is not the case.



We even see that the test error decreases after the training error is zero!

We will describe adaBoost in a diffrent way that will explain this.

Remember $F(x) = \sum \alpha_i h_i(x)$ and $H(x) = sign(F(x))$. We defined an exponential loss that bounds the $0-1$ loss, $\exp(-yF(x))$.

We will see that adaBoost is a greedy algorithm to minize the exponential loss.

This leads to large margins, and that implies generalization (even with large VC dimension).

Algorithm `Greedy exponential loss`

Input: training set $S = (x_1, y_1), ..., (x_m, y_m)$.
Initialize: $F_0(x) = 0$
for t=1,...,T **do**
  Chose $h_t \in B$ and α_t to minimize
  $\frac{1}{m}\sum_{i=1}^{m} \exp(-y_i(F_{t-1}(x_i) + \alpha_t h_t(x_i))$
  Update: $F_t = F_{t-1} + \alpha_t h_t$.
end for
return F_T

We will show that this algorithm is indeed adaBoost.

Proof:

$$\frac{1}{m}\sum_{i=1}^{m}\exp(-y_iF_{t-1}(x_i)+\alpha_t h_t(x_i)) =$$

$$\frac{1}{m}\sum_{i=1}^{m}\exp(-y_iF_{t-1}(x_i))\exp(-y_i\alpha_t h_i(x)) \propto \sum_{i=1}^{m}\mathbf{D}^t(i)\exp(-y_i\alpha_t h_i(x))$$

Which is Z_t. For the optimal h_t with error ϵ_t we get $Z_t = e^{-\alpha_t}(1-\epsilon_t) + e^{\alpha_t}\epsilon_t$ which is optimized by the α_t chosen by adaBoost to be equal $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$.

We just need to show that we have picked the h_t adaBoost returns.

This is easy as Z_t is decreasing for $0 < \epsilon_t < 1/2$, so it is minimized by minimizing ϵ_t which is exactly what adaBoost does.

Looking at the exponential error, we see that the adaBoost will try to maximize the margins.

We will prove a genralization bound for large margins. First a quick reminder on Rademacher complexity

$$R(\mathcal{F} \circ S) = \frac{1}{m}\mathbb{E}_{\sigma \sim \{\pm 1\}^m}\left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{m} \sigma_i f(z_i)\right].$$

We proved (more or less) that if $\mathcal{F}$ is a family of functions into $[-1, 1]$ then with probability greater or equal to $\geq 1 - \delta$ we have *for all* $f \in \mathcal{F}$,

$$\mathbb{E}_{z \sim \mathcal{D}}[f(z)] \leq \mathbb{E}_{z \sim S}[f(z)] + 2R(\mathcal{F} \circ S) + \sqrt{\frac{2\ln(2/\delta)}{m}} \tag{5}$$

Assume the weak classifiers are in a space B with VC dimension d. AdaBoost returns $H(x) = sign(\sum \alpha_i h_i(x))$, with $\alpha_i > 0$.

We can normalize $a_i = \alpha_i / \sum \alpha_i$, and define $f(x) = \sum a_i h_i(x)$. Notice $f(x) \in [-1, 1]$, $sign(f(x)) = H(x)$ and $f \in conv(B)$.

Theorem 3.1

$$P_{\mathcal{D}}[yf(x) \leq 0] \leq P_S[yf(x) \leq \theta] + \frac{2}{\theta} \cdot \sqrt{\frac{2d \ln(em/d)}{m}} + \sqrt{\frac{2 \ln(2/\delta)}{m}}$$

Proof: Define an auxiliary function ϕ

$$\phi(x) = \begin{cases} 1 & : x < 1 \\ 1 - x/\theta & : 0 \leq x \leq \theta \\ 0 & : x > \theta \end{cases}$$

It is easy to see that $\mathbb{1}[yf(x) \leq 0] \leq \phi(yf(x)) \leq \mathbb{1}[yf(x) \leq \theta]$.

This means $P_{\mathcal{D}}(yf(x) \leq 0) \leq \mathbb{E}_{\mathcal{D}}[\phi(yf(x))]$ and $\mathbb{E}_S[\phi(yf(x))] \leq P_S(yf(x) \leq \theta)$

So to prove the theorem it is enough to show $R(\phi \circ \mathcal{F} \circ S) \leq \frac{2}{\theta} \cdot \sqrt{\frac{2d \ln(em/d)}{m}}$, but this is trivial using the fact that $\mathcal{F} \circ S = conv(B \circ S)$ and ϕ is $1/\theta$-Lipschitz.