CS 7545: Machine Learning Theory **Fall 2021**

Lecture 12: Boosting and SVM

Instructor: Santosh Vempala *Lecture date: 10/25-27/2021*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

12.1 Boosting

Let $\mathcal{H}$ be a hypothesis class of Boolean valued functions.

Definition 12.1 (Weak Learner)**.** *A hypothesis h is a weak learner with $\gamma > 0$ if*

$$\Pr_D(h(x) = \ell(x)) \geq 1/2 + \gamma$$

where $\ell(x)$ is the true labeling function over D.

Remark 12.2. *Random guessing gives a 1/2 correctness. The performance of a weaker learner is slightly better than random guessing.*

Definition 12.3 (Strong Learner)**.** *For any $\epsilon > 0$, a hypothesis h is a strong learner if*

$$\Pr_D(h(x) = \ell(x)) \leq \epsilon$$

where $\ell(x)$ is the true labeling function over D.

Suppose we know how to get a weak learner $h \in \mathcal{H}$ for any distribution D, can we create a strong learner?

Algorithm 1 Boosting

Initialize $w_i \leftarrow 1$ for each sample $x_i \in S$
for $t = 1, \ldots, T$
  $h_i \leftarrow$ the concept that correctly classifies $1/2 + \gamma$ fraction of the current total weight
  Increase the weight of each example mis-classified by h_i by a factor of $\frac{\frac{1}{2}+\gamma}{\frac{1}{2}-\gamma}$
Output $\hat{h} = \mathrm{MAJ}(h_1, h_2, \ldots, h_T)$

Bound on number of Mistakes:

- Let the number of error made by the final majority hypothesis be m.
- If $\mathrm{MAJ}(h_1, h_2, \ldots, h_T)$ misclassifies x_i, at least $T/2$ h_i's must misclassify x_i. So,

$$w_i \geq \left(\frac{\frac{1}{2}+\gamma}{\frac{1}{2}-\gamma}\right)^{\frac{T}{2}} = \left(\frac{1+2\gamma}{1-2\gamma}\right)^{\frac{T}{2}}. \tag{12.1}$$

- Let W_t = total weight at the t-th step. Then, $W_0 = n$ and

$$W_{t+1} \leq W_t \cdot \left[\left(\frac{1}{2} - \gamma\right) \cdot \left(\frac{\frac{1}{2}+\gamma}{\frac{1}{2}-\gamma}\right) + \left(\frac{1}{2} + \gamma\right)\right] = W_t \cdot (1 + 2\gamma). \tag{12.2}$$

- From (12.1) and (12.2), $m\left(\frac{1+2\gamma}{1-2\gamma}\right)^{\frac{T}{2}} \leq W_T \leq n(1+2\gamma)^T$ which gives $m \leq (1-4\gamma^2)^{\frac{T}{2}} n$.
- For $T = \frac{\ln n}{\gamma^2}$, $m < 1 \Rightarrow m = 0$.

So, $\hat{h}(x_i) = \ell(x_i)$ for all $i \in [n]$. To guarantee (ϵ, δ)-PAC learning, we need

$$n = O\left(\frac{1}{\epsilon}\left(\text{VC-dim}(\text{MAJ}_k(\mathcal{H}))\log\frac{1}{\epsilon} + \log\frac{1}{\delta}\right)\right),$$

where $\text{MAJ}_k(\mathcal{H})$ is the hypothesis class of majority of k concepts from $\mathcal{H}$.

Theorem 12.4. *If a hypothesis class $\mathcal{H}$ has VC-dim d, then majority of k concepts from $\mathcal{H}$ has VC-dim at most $2kd\log(kd)$.*

Proof. The number of ways concepts in $\mathcal{H}$ can label m points is at most m^d. The number of ways majority of k concepts in $\mathcal{H}$ can label m points is at most m^{kd}. Let $\hat{d}$ be the VC-dim of class of majority of k concepts from $\mathcal{H}$. Then, $2^{\hat{d}} \leq \hat{d}^{kd} \Rightarrow \hat{d} \leq 2kd\log(kd)$. □

Corollary 12.5. $n = O\left(\frac{1}{\epsilon}\left(Td\log(Td)\log\frac{1}{\epsilon} + \log\frac{1}{\delta}\right)\right)$ *for* $T = \frac{\ln n}{\gamma^2}$ *examples imply (ϵ, δ)-PAC learning.*

12.2 Support Vector Machines

Given data points $\{x_i\}$, we want to find a separator w that minimizes the Hinge Loss, which is defined as

$$\begin{aligned} &\min \sum_{i=1}^{m} \epsilon_i \\ \text{s.t.}\quad & w \cdot x_i \geq 1 - \epsilon_i \ \text{ if } \ell(x_i) = 1 \\ & w \cdot x_i \leq -1 + \epsilon_i \ \text{ if } \ell(x_i) = -1 \\ & \epsilon_i \geq 0 \end{aligned}$$

If $OPT = 0$, then there exists a perfect classifier. However, the Hinge Loss does not guarantee anything about the margin of the classifier. In practice, it might be preferable to have a classifier with a small amount of error but a large margin rather than a perfect classifier with a small margin. Support Vector Machines deal with this issue by regularizing the Hinge Loss with margin. Recall that the margin for a vector w^* is defined as $\gamma = \min_x \frac{|w^* \cdot x|}{\|w^*\|^2}$. This implies $\|w^*\| = \min_x \frac{|w^* \cdot x|}{\gamma^2} \leq \frac{1}{\gamma^2}$. The Support Vector Machine solves the following convex optimization problem

$$\begin{aligned} &\min \|w\|^2 + c\sum_{i=1}^{m} \epsilon_i \\ \text{s.t.}\quad & w \cdot x_i \geq 1 - \epsilon_i \ \text{ if } \ell(x_i) = 1 \\ & w \cdot x_i \leq -1 + \epsilon_i \ \text{ if } \ell(x_i) = -1 \\ & \epsilon_i \geq 0. \end{aligned}$$

Here c is the relative weight of Hinge Loss. The choice of c depends on the data or the application.

Theorem 12.6. *The number of mistakes made by the Perceptron algorithm is*

$$\#\textit{mistakes} \leq \min_w \left(\frac{1}{\gamma_w^2} + 2 \cdot \textit{(Hinge Loss of } w\textit{)}\right)$$

Proof. Consider the potential $w \cdot w^*$. When the algorithm makes a mistake,

$$w \cdot w^* \leftarrow w \cdot w^* + \ell(x_i)(x_i \cdot w^*) \geq w \cdot w^* + 1 - \epsilon_i,$$

where $\ell(x_i)(w^* \cdot x) \geq 1 - \epsilon_i$ for all i. After M mistakes,

$$w \cdot w^* \geq M - \sum_i \epsilon_i \geq M - L,$$

where L is the Hinge Loss of w^*. When the algorithm makes a mistake,

$$w \cdot w \leftarrow w \cdot w + (x_i \cdot x_i) + 2\ell(x_i)(w \cdot x_i) \leq w \cdot w + 1.$$

After M mistakes, $\|w\|^2 \leq M$. Using Cauchy Schwarz, $|w \cdot w^*| \leq \|w\| \cdot \|w^*\|$, which implies $M - L \leq \|w^*\| \sqrt{M}$. On squaring both sides,

$$(M - L)^2 \leq \|w^*\|^2 M \Rightarrow M \leq \|w^*\|^2 + 2L \leq \frac{1}{\gamma_{w^*}^2} + 2L.$$

$\square$

12.3 Random Projections

Given samples in $\mathbb{R}^d$, consider the random projection matrix $R : \mathbb{R}^d \to \mathbb{R}^k$, where each entry r_{ij} is sampled independently from $N(0, 1/\sqrt{k})$. Let $x' = R^\top x$. Then $\mathbb{E}[\|x'\|^2] = \|x\|^2$.

Theorem 12.7. *For a random projection matrix $R : \mathbb{R}^d \to \mathbb{R}^k$ and $x \in \mathbb{R}^d$,*

$$\Pr(|\, \left\|R^\top x\right\|^2 - \|x\|^2 \,| \geq \epsilon \|x\|^2) \leq 2e^{-\frac{(\epsilon^2 - \epsilon^3)k}{4}}$$

Therefore, $k = O\left(\frac{1}{\epsilon^2} \log \frac{m}{\delta}\right)$ preserves the lengths of m vectors.

Theorem 12.8. *For a random projection matrix $R : \mathbb{R}^d \to \mathbb{R}^k$ and $x, y \in \mathbb{R}^d$,*

$$\Pr(|(R^\top x) \cdot (R^\top y) - x \cdot y| \geq \epsilon \|x\| \|y\|) \leq 2e^{-c\epsilon^2 k}.$$

Consider the setting when we are trying to learn a halfspace with margin γ in $\mathbb{R}^d$. If we first randomly project to $\mathbb{R}^k$ for $k = O\left(\frac{1}{\gamma^2} \log \frac{m}{\delta}\right)$, we get a margin on at least $\gamma/2$ w.h.p., and to learn the halfspace in $\mathbb{R}^k$, we need only $m = O(\frac{k}{\epsilon} \log \frac{1}{\epsilon} + \frac{1}{\epsilon} \log \frac{1}{\delta})$ samples. Therefore,

$$k = O\left(\frac{1}{\gamma^2} \log \frac{1}{\gamma\epsilon\delta}\right), \quad \text{and}$$
$$m = O\left(\frac{1}{\epsilon\gamma^2} \log \frac{1}{\gamma\epsilon\delta} \log \frac{1}{\epsilon}\right).$$