6.842 Randomness and Computation Nov 27, 2017

Lecture 22

Lecturer: Ronitt Rubinfeld *Scribe: Suhas S Kowshik*

1 Weak vs Strong Learning

In this and the subsequent lecture we show how weak learning implies strong learning. First we recall the definitions:

Definition 1 *Algorithm $\mathcal{A}$* strongly *PAC learns a concept class $\mathcal{C}$ if $\forall c \in \mathcal{C}$, for every distribution $\mathcal{D}$ on the input, $\forall \epsilon, \delta > 0$, given examples of c according to $\mathcal{D}$, with probability at least $1-\delta$, $\mathcal{A}$ outputs hypothesis h such that $\mathbb{P}_{\mathcal{D}}[h(X) \neq c(X)] \leq \epsilon$.*

Definition 2 *Algorithm $\mathcal{A}$* weakly *PAC learns a concept class $\mathcal{C}$ if $\forall c \in \mathcal{C}$, for every distribution $\mathcal{D}$ on the input, $\exists \gamma > 0$ such that $\forall \delta > 0$, given examples of c according to $\mathcal{D}$, with probability at least $1-\delta$, $\mathcal{A}$ outputs hypothesis h such that $\mathbb{P}_{\mathcal{D}}[h(X) = c(X)] \geq \frac{1}{2} + \frac{\gamma}{2}$.*

Notice that in the definition of weak learning, we do not have control over the parameter γ. Hence it is weaker than strong learning where we can learn to any accuracy $(1-\epsilon)$.

Our goal is to prove the following theorem:

Theorem 3 *If concept class $\mathcal{C}$ can be weakly learned on any $\mathcal{D}$ then it can be strongly learned.*

To prove this theorem, we introduce a technique called boosting.

2 Boosting

2.1 Intuition

Suppose we have a weak learner $\mathcal{A}$ for a concept class $\mathcal{C}$. First idea could be to use this learner multiple times on the same distribution $\mathcal{D}$, and then take majority of the hypotheses predicted each time. But repeated trials can only help boost confidence δ, but not the accuracy. For example, if each time, the hypothesis is same h, then repeating and taking majority wouldn't help.

The next idea could be to do repetition but an intelligent one! So what we could is to run the weak learner on those samples where the previous hypotheses failed to predict correctly. More formally, if the hypothesis in first stage is c_1 based on the samples $\{(x_i, c(x_i))\}_{i=1}^{m}$ from $\mathcal{D}$ (where m is the samples needed by $\mathcal{A}$), then we query more samples $(x_{m+1}, c(x_{m+1})), \ldots$ but use only those samples for which $c(x_j) \neq c_1(x_j)$. Thus we *filter* out the samples, and this induces a new distribution $\mathcal{D}_1$ on the input. We use these filtered samples to get an hypothesis c_2. But then how do we continue and at the end, what is our hypothesis?

The boosting technique use majority of the hypotheses for filtering at each stage, and the final output is also the majority of all the hypotheses seen so far. But there are some modifications required to the filtering process which is discussed later.

2.2 Algorithm

Suppose we are given samples from $\mathcal{D}$ labelled according to c and weak learner (WL) $\mathcal{A}$

- Stage 0: $\mathcal{D}_0 \leftarrow \mathcal{D}$. Run WL on $\mathcal{D}_0$ to generate hypothesis c_1 such that $\mathbb{P}_{\mathcal{D}_0}[c(x) = c_1(x)] \geq \frac{1}{2} + \frac{\gamma}{2}$.

- For stage $i = 1...T$, where $T = O(\frac{1}{\gamma^2\epsilon^2})$
  - Stop if $Maj(c_1, .., .c_i)$ is correct on $1-\epsilon/2$ fraction of the input w.r.t $\mathcal{D}$, and output $Maj(c_1, .., .c_i)$. This can be done by taking samples from $\mathcal{D}$ and checking how many fail. We use $\epsilon/2$ here to account for error due to sampling.
  - Using $c_1, c_2, ..., c_i$, construct $\mathcal{D}_i$ using some "filtering" procedure.
  - Run WL filtered samples (according to $\mathcal{D}_i$) to get hypothesis c_{i+1} such that $\mathbb{P}_{\mathcal{D}_i}[c(x) = c_{i+1}(x)] \geq \frac{1}{2} + \frac{\gamma}{2}$
- Output $C^* = Maj(c_1, ..., c_T)$.

Next, we describe the filtering procedure.

2.3 Filtering

Given $c_1, ..., c_i$ and a new labelled sample $(x, c(x))$:

- If $Maj(c_1, ..., c_i)(x) \neq c(x)$ then keep $(x, c(x))$.
- Let $\#right$ be the number of correct predictions of x among $c_1, ..., c_i$ and let $\#wrong$ be the number of incorrect predictions. If $\#right - \#wrong \geq \frac{1}{\gamma\epsilon}$ then discard x.
- Else if $\#right - \#wrong = \frac{\alpha}{\gamma\epsilon}$ for $0 < \alpha < 1$, then keep x with probability $1 - \alpha$.

As a remark, we must ensure that the WL gets enough samples after filtering. That is, the sample complexity shouldn't blowup. But notice that we enter the filtering stage only if $\mathbb{P}_{\mathcal{D}}[c(x) \neq Maj(c_1, ..., c_i)(x)] > \epsilon$. Hence atleast one out of $O(1/\epsilon)$ samples will pass through the filter with high probability.

In the next section we introduce some notations which are useful in proving the correctness of the boosting algorithm.

3 Notations

For each i, define the correctness of a prediction c_i on x as

$$R_{c_i}(x) = \begin{cases} +1 & \text{if } c_i(x) = c(x) \\ -1 & \text{if } c_i(x) \neq c(x). \end{cases}$$

Let $N_i(x) = \#right - \#wrong$ after i steps:

$$N_i(x) = \sum_{1 \leq j \leq i} R_{c_i}(x).$$

Define a measure $M_i(x)$ as

$$M_i(x) = \begin{cases} 1 & \text{if } N_i(x) \leq 0 \\ 0 & \text{if } N_i(x) \geq \frac{1}{\gamma\epsilon} \\ 1 - \epsilon\gamma N_i(x) & \text{otherwise.} \end{cases}$$

Let distribution $\mathcal{D}_{M_i}$ be defined as

$$\mathcal{D}_{M_i}(x) = \frac{M_i(x)\mathcal{D}(x)}{\sum_{y \in \{+1,-1\}^n} M_i(y)\mathcal{D}(y)}.$$

So, this distribution coincides with the distribution $\mathcal{D}_i$ described in the algorithm.

From now on, we assume that $\mathcal{D}$ is the uniform distribution. Hence

$$\mathcal{D}_{M_i}(x) = \frac{M_i(x)}{\sum_{y \in \{+1,-1\}^n} M_i(y)}.$$

We also define the advantage of a prediction $\tilde{c}$ over M_i:

$$Adv_{\tilde{c}}(M_i) = \sum_x R_{\tilde{c}}(x) M_i(x).$$

Hence we have

$$\mathbb{P}_{\mathcal{D}_i}[\tilde{c}(x) = c(x)] = \frac{1}{2} + \frac{Adv_{\tilde{c}}(M_i)}{2 \sum_x M_i(x)}.$$

We will continue the proof in the next lecture.