Alin Tomescu
6.867 Machine learning | Week 6, Tuesday, October 10th, 2013| Lecture 11

Lecture 11

Understanding generalizations, what kind of guarantees can we give about a learning algorithm?

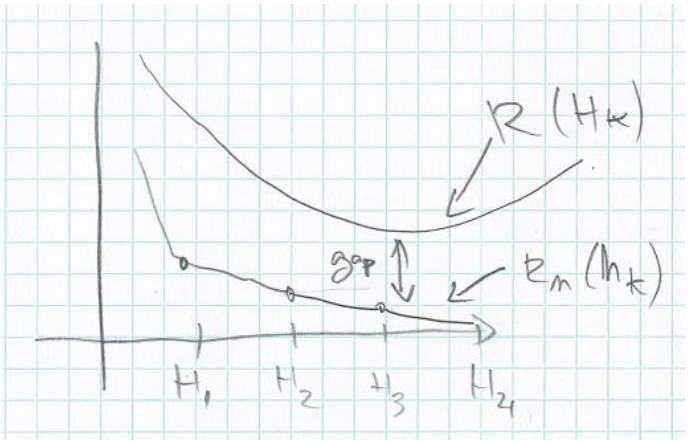
Today, we will work with the training error:

$$R_n(h) = \frac{1}{n}\sum_{i=1}^{n} Loss_{0,1}\left(y^{(i)}h\left(x^{(i)}\right)\right), \left(x^{(i)}, y^{(i)}\right) \sim p^*$$

We wish to optimize the generalization error:

$$R(h) = E_{(x,y)\sim p^*}\{Loss_{0,1}(yh(x))\}$$

$$\mathcal{H} = \text{set of classifiers}$$



We are looking for an upper bound on $R(h)$, such that $R(h) \leq R_n(h) + \varepsilon$

The gap depends on the number of training examples. (because you can better train the classifier with more examples?)

Cases:

(1) $|\mathcal{H}| < \infty, \mathcal{H} = \{h_1, \dots, h_k\}$, realizable, which means $\exists h^* \in \mathcal{H}, R(h^*) = 0$. What guarantees can we give in this case?
(2) $|\mathcal{H}| < \infty, \mathcal{H} = \{h_1, \dots, h_k\}$, but not realizable
(3) $|\mathcal{H}| = \infty$ for example, the set of linear classifiers is such an $\mathcal{H}$
(4) No longer pick one single classifier, but pick a distribution over the classifiers $h \in \mathcal{H}$
    a. Have a "prior" $P(h)$, and I will select a "posterior" $Q(h)$
    b. In this case, the training error corresponds to taking an expected value of the generalization error of the classifier over the distribution that I choose
        i. $E_{h\sim Q}\{R_n(h)\}$
        ii. $E_{h\sim Q}\{R(h)\}$

Alin Tomescu

6.867 Machine learning | Week 6, Tuesday, October 10th, 2013| Lecture 11

Case 1: $|\mathcal{H}| < \infty, \exists h^* \in \mathcal{H}, R(h^*) = 0$

The types of guarantees we hope to have in case 1, is that the generalization error is lower than the training error. With high probability, we want:

$$\Pr\left(R\left(\hat{h}\right) \leq R_n\left(\hat{h}\right) + \varepsilon\right) \geq 1 - \delta$$

$$\varepsilon = \varepsilon(\mathcal{H}, n, \delta)$$

If the problem is realizable, then the training error is

$$R_n\left(\hat{h}\right) = \min_{h \in \mathcal{H}} R_n(h) = 0$$

Since, we know there exists a classifier that generalizes perfectly, that classifier will also have to train perfectly (with no - training errors).

$$\Pr\left(R\left(\hat{h}\right) \leq R_n\left(\hat{h}\right) + \varepsilon\right) \geq 1 - \delta \Leftrightarrow \Pr(\exists h \in \mathcal{H}\ R_n(h) = 0, R(h) > \varepsilon) < \delta$$

$$(R_n\left(\hat{h}\right) = \min_{h \in \mathcal{H}} R_n(h) = 0)$$

We do not consider classifiers that have non-zero training error since we only care about the ones that generalize perfectly. The probability that there exists a classifier that violates the first probability ($\Pr\left(R\left(\hat{h}\right) \leq R_n\left(\hat{h}\right) + \varepsilon\right) \geq 1 - \delta$) is less that δ.

Let:

$$R(h) = \varepsilon_h$$

Let's pick h such that $\varepsilon_h > \varepsilon$. If I pick some classifier that does not generalizes as well as I want what is the probability that it survives my screening process.

$$\Pr(R_n(h) = 0) = (1 - \varepsilon_h)^n \leq (1 - \varepsilon)^n$$

This means that the generalization error is exactly the probability that I would make an error on a randomly chosen training example. There are n such examples.

Union bound: $\Pr(A_1\ or\ A_2\ or\ A_3\ or\ \ldots) \leq \Pr(A_1) + \Pr(A_2) + \cdots + \Pr(A_3)$

$$\Pr(\exists h \in \mathcal{H}\ R_n(h) = 0, R(h) > \varepsilon) \leq \sum_{h \in \mathcal{H}, \varepsilon_h > \epsilon} \Pr(R_n(h) = 0)$$

But, $\Pr(R_n(h) = 0) \leq (1 - \varepsilon)^n$, so:

$$\Pr(\exists h \in \mathcal{H}\ R_n(h) = 0, R(h) > \varepsilon) \leq |\mathcal{H}|(1 - \varepsilon)^n = \delta$$

So, with probability at least $1 - \delta$:

$$R\left(\hat{h}\right) \leq R_n\left(\hat{h}\right) + \varepsilon(\mathcal{H}, n, \delta), \varepsilon = \frac{\log|\mathcal{H}| + \log\frac{1}{\delta}}{n}, when\ 1 - \varepsilon \leq e^{-n\varepsilon}$$

Alin Tomescu

The gap **goes down** as a function of the training examples. The **gap increases logarithmically in terms of the size of sets of classifiers**.

Case 2: $|\mathcal{H}| < \infty, \forall h^* \in \mathcal{H}, R(h^*) \neq 0$

$$\Pr\left(R(\hat{h}) \leq R_n(\hat{h}) + \varepsilon\right) \geq 1 - \delta$$

Now there's no h that generalizes perfectly, so this is harder.

If we prove something stronger, then the above statement is also true.

$$\Pr(\forall h \in \mathcal{H}, R(h) \leq R_n(h) + \varepsilon) \geq 1 - \delta$$

As a result the ε will be a little bit larger than we would like.

$$\Leftrightarrow \Pr(\exists h \in \mathcal{H}, R(h) > R_n(h) + \varepsilon) \leq \delta$$

We can use the union bound:

$$\Pr(\exists h \in \mathcal{H}, R(h) > R_n(h) + \varepsilon) \leq \sum_{h \in \mathcal{H}} \Pr(R(h) > R_n(h) + \varepsilon)$$

What is $\Pr(R(h) > R_n(h) + \varepsilon)$?

We can define an R.V. $S_i = Loss_{0,1}\left(y^{(i)}h\left(x^{(i)}\right)\right) =$ the loss on the i^{th} training example.

What is $E\{S_i\}$?

$$E\{S_i\} = E_{(x^{(i)},y^{(i)})\sim p^*}\left\{Loss_{0,1}\left(y^{(i)}h\left(x^{(i)}\right)\right)\right\} = E_{(x,y)\sim p^*}\{Loss_{0,1}(yh(x))\} = R(h)$$

So the expected value of the training error if you do not train the classifier is exactly the generalization error. But it is not the generalization error when you adjust h based on the training set.

$$S = S_i$$

$$\Pr\left(E\{S\} > \frac{1}{n}\sum_{i=1}^{n} S_i + \varepsilon\right) \leq e^{-2n\varepsilon^2} \ (by\ Chernoff\ bound)$$

So,

$$\Pr(\exists h \in \mathcal{H}, R(h) > R_n(h) + \varepsilon) \leq |\mathcal{H}|e^{-2n\varepsilon^2} = \delta$$

$$\varepsilon = \sqrt{\frac{\log|\mathcal{H}| + \log\left(\frac{1}{\delta}\right)}{2n}}$$

TODO: Make sure this is a $2n$ and not a 2

With probability at least $1 - \delta$, for all classifiers $h \in \mathcal{H}, R(h) \leq R_n(h) + \varepsilon, \varepsilon = \sqrt{\frac{\log|\mathcal{H}| + \log\left(\frac{1}{\delta}\right)}{2n}}$

Alin Tomescu

This result is poorer, since the gap is smaller.

Case 3: $|\mathcal{H}| = \infty$

$|\mathcal{H}| = \infty$ (example: linear classifiers).

Many of the classifier choices perform the same. So somehow we have to collapse these togethere.

$$x^{(1)}, \dots, x^{(n)}$$

Pick $h_1 \in \mathcal{H}$, that predicts $+, -, \dots +$

Pick $h_2 \in \mathcal{H}$, that predicts $-, -, \dots +$

Pick $h_3 \in \mathcal{H}$, that predicts $+, -, \dots +$

In some sense h_1 and h_3 are the same (roughly equal), because they classify the training set the same.

It turns out there is a finite # of distinct labelings. Let's call this number $\mathcal{N}_{\mathcal{H}}\left(x^{(1)}, \dots, x^{(n)}\right)$.

$$\mathcal{N}_{\mathcal{H}}(n) = \max_{x^{(1)},\dots,x^{(n)}} \mathcal{N}_{\mathcal{H}}\left(x^{(1)}, \dots, x^{(n)}\right)$$

This is known as the **growth function**. "Find the set of examples that maximize the # of distinct labelings".

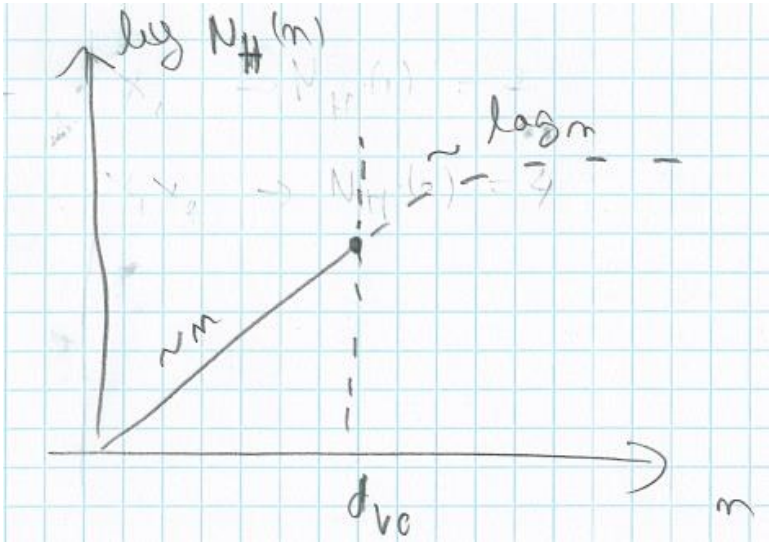
Let's take an example, like linear classifiers in 2D, and see how this behaves.

$$x_1 \to \mathcal{N}_{\mathcal{H}}(1) = 2^1$$

$$x_1, x_2 \to \mathcal{N}_{\mathcal{H}}(2) = 2^2$$

$$x_1, x_2, x_3 \to \mathcal{N}_{\mathcal{H}}(3) = 2^3$$

$x_1, x_2, x_3, x_4 \to \mathcal{N}_{\mathcal{H}}(4) = 14 < 2^4$ (because we cannot separate the XOR function)



$$d_{VC} = \max\{h : \mathcal{N}_{\mathcal{H}}(h) = 2^n\} = max\ \#\ of\ points\ for\ which\ the\ growth\ function\ is\ exactly\ 2^n$$

Alin Tomescu
6.867 Machine learning | Week 6, Tuesday, October 10th, 2013| Lecture 11

Definition: A classifier can *shatter* a set of points when the classifier can generate all instances of possible labelings over the points. Or, if the classifier can classify the points correctly independent of the labeling.