10-704: Information Processing and Learning **Spring 2015**

Lecture 11: Feb 17

Lecturer: Aarti Singh *Scribes: Xuanchong Li*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

11.1 Empirical Risk Minimization

In many machine learning task, the task is to minimizing the risk.

$$R(f) = \mathbb{E}[\ell(f(X), Y)] \tag{11.1}$$

where ℓ is a loss function of interest. In practice, we compute the empirical risk.

$$\hat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(X_i), Y_i) \tag{11.2}$$

We choose the $\hat{f}$ that minimizes the empirical risk.

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \hat{R}(f) \tag{11.3}$$

To justify this empirical risk minimization (ERM) method, we need to know how similar the $R(f)$ and $\hat{R}$ are. According to PAC theory, if ℓ is bounded, with the probability at least $1 - \delta(f)$, we have the following bound for a given $f \in \mathcal{F}$:

$$R(f) \leq \hat{R}_n(f) + \sqrt{\frac{\log \frac{1}{\delta(f)}}{2n}} \tag{11.4}$$

When $\mathcal{F}$ is finite, we can take $\delta(f) = \delta \frac{1}{|\mathcal{F}|}$. Then, with probability $\geq 1 - \delta$,

$$R(f) \leq \hat{R}_n(f) + \sqrt{\frac{\log |\mathcal{F}| + \log \frac{1}{\delta}}{2n}}, \forall f \in \mathcal{F} \tag{11.5}$$

where $\log |\mathcal{F}|$ can be considered as the number of bits needed to encode $\mathcal{F}$.
For ERM,

$$E[\hat{R}(f)] \leq \min_{f \in \mathcal{F}} R(f) + \sqrt{\frac{\log |\mathcal{F}| + \log \frac{1}{\delta}}{2n}} + \delta, \forall f \in \mathcal{F} \tag{11.6}$$

When $\mathcal{F}$ is countably infinite, we can use the prefix code. Let $c(f)$ denote the prefix code length for encoding f. A more complex model needs longer code to encode. According to Kraft's inequality,

$$\sum_{f \in \mathcal{F}} D^{-c(f)} \leq 1 \tag{11.7}$$

when using D-ary code. We can take $\delta(f) = \delta D^{-c(f)}$. Then the bound is, with probability $\geq 1-\delta$, for all $f \in \mathcal{F}$,

$$R(f) \leq \hat{R}_n(f) + \sqrt{\frac{c(f) + \log \frac{1}{\delta}}{2n}}, D = e \tag{11.8}$$

It implies the empirical risk is closer to true risk when f is simpler (needs fewer bits to encode).

11.2 Complexity Regularized Empirical Risk Minimization

To achieve better estimation of the true risk, we should minimize both the empirical risk and complexity, instead of only minimizing the empirical risk.

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \{\hat{R}_n(f) + \epsilon(f)\} \tag{11.9}$$

where $\epsilon(f) = \sqrt{\frac{c(f)+\log \frac{1}{\delta}}{2n}}$. With probability $\geq 1-\delta$, we have the following bound on $R(\hat{f})$

$$\begin{aligned} R(\hat{f}) &\leq \hat{R}(\hat{f}) + \epsilon(\hat{f}) && (11.10)\\ &\leq \hat{R}(f) + \epsilon(f), \forall f \in \mathcal{F} && (11.11) \end{aligned}$$

Let Ω denote the event on which we have bounded the deviation of true and empirical risks for all $f \in \mathcal{F}$, which means $P(\Omega) \geq 1-\delta$. Also, let Ω^c denote the complement set of Ω, then the expectation of $R(\hat{f})$ is also bounded as follows.

$$E[R(\hat{f})] \leq E[R(\hat{f})|\Omega]P(\Omega) + E[R(\hat{f})|\Omega^c]P(\Omega^c) \tag{11.12}$$

Since

$$P(\Omega) \leq 1$$

$$P(\Omega^c) \leq \delta$$

and

$$E[R(\hat{f})|\Omega^c] = O(1) \text{ (loss function is bounded)}$$

we have

$$E[R(\hat{f})] \leq R(f) + \epsilon(f) + O(\delta) \tag{11.13}$$

Since this is true for all $f \in \mathcal{F}$, we get the following which states that the complexity penalized ERM balances both the risk and complexity.

$$E[R(\hat{f})] \leq \min_{f \in \mathcal{F}} \{R(f) + \epsilon(f)\} + O(\delta) \tag{11.14}$$

Let R^* denotes the risk of the best f, then for any $\delta \in (0,1)$

$$E[R(\hat{f})] - R^* \leq \min_{f \in \mathcal{F}} \{(R(f) - R^*) + \epsilon(f)\} + O(\delta) \tag{11.15}$$

where $R(f) - R^*$ is the *approximation error* which tells us how well a function approximates the optimal predictor, and $\epsilon(f)$ the *estimation error* which tells us how the true and empirical risks deviate for f. Typically, the simpler f is, the smaller the estimation error but the larger the approximation error. The complexity penalized ERM picks a function that balances these two errors.

Next, we will apply this to classification (which involves the 0/1 loss function which is bounded by 1).

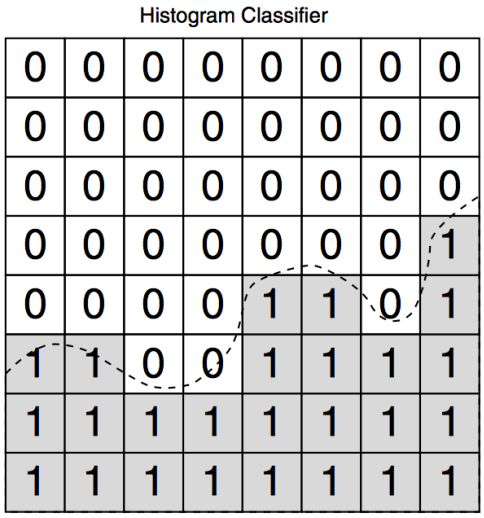

Figure 11.1: Histogram Classifier

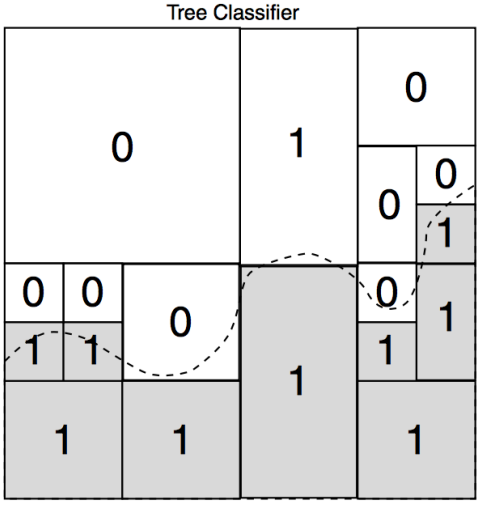

Figure 11.2: Tree Classifier

11.3 Histogram Classifiers

The histogram classification, shown in Figure 11.2, is similar to histogram density estimation. It divides the domain into bins. For the points inside each bin, it assigns a label. For example, it can use the majority vote to decide the label in each bin.

To use the complexity penalized ERM, we need to know how to do the prefix code for histogram classifiers.

Let $\mathcal{F}_m$ denote the class of histogram classifiers with m bins. Suppose there are two classes 0 and 1. Then $|\mathcal{F}_m| = 2^m$. To encode 2^m different classifiers, we need $\log 2^m = m$ bits. In addition, we need to encode the integer m denoting the histogram resolution, which needs $\log m$ bits. Therefore, we need $c(f) = O(m)$ bits to encode $f \in \mathcal{F}_m$. We will consider the countably infinite class $\mathcal{F} = \cup_m \mathcal{F}_m$.

Plugging this in the earlier expression, we have the complexity penalized ERM (CRM) classifier.

$$\hat{f}_{CRM} = \min_{f \in \mathcal{F}} \{\hat{R}(f) + \sqrt{\frac{O(m) + \log \frac{1}{\delta}}{2n}}\} = \min_m \{\min_{f \in \mathcal{F}_m} \hat{R}(f) + \sqrt{\frac{O(m) + \log \frac{1}{\delta}}{2n}}\} \tag{11.16}$$

The error bound is

$$E[R(\hat{f}_{CRM})] \quad \leq \quad \min_m \{\min_{f \in \mathcal{F}_m} R(f) + \sqrt{\frac{O(m) + \log \frac{1}{\delta}}{2n}}\} + \delta \tag{11.17}$$

Thus, the complexity penalized procedure also performs model selection (pick the best m) for histogram classifiers automatically.

11.4 Decision Trees Classifiers

Decision trees classifiers are related to histogram classifiers. Rather than only using uniform bin size in histogram classifiers, decision trees classifiers allow different bin sizes.

For example, the axis parallel tree splits the space in the axis parallel way -alternating between vertical and horizontal splits. Here we only consider the dyadic split, which means the split can only happen on the middle value. Figure 11.4 shows an example of recursive dyadic partition tree growing.

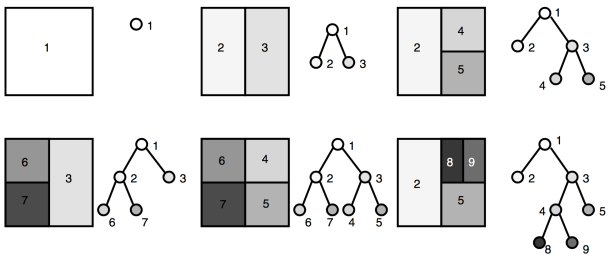

Figure 11.3: Example of Recursive Dyadic Partition (RDP) growing

Now wee need to encode such dyadic decision trees classifiers. In specific, we need to encode the structure of the tree and the label on the leaves. Let $\mathcal{F}_K$ denotes the class of decision trees with K leaves. We will do CRM over the countably infinite class $\cup_K \mathcal{F}_K$.

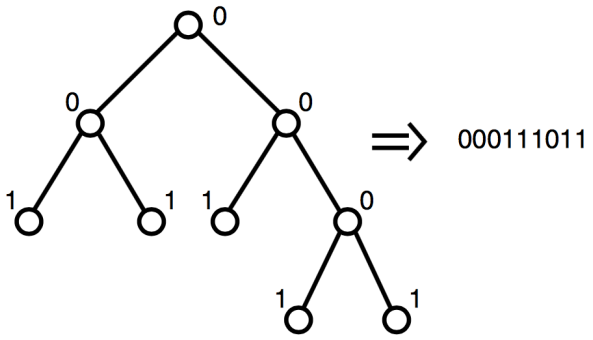

Figure 11.4: Example of a tree and corresponding prefix code.

For encoding tree structure, we can assign 0 for internal nodes and 1 for leaves. Then we traverse the tree top-to-bottom and in left-to-right order to produce the code. An example is shown in Figure 11.4. So the number of bits needed to encode the tree structure is equal to the number of nodes. For a tree with K leaves, there are $2K-1$ nodes. Hence we need $2K-1$ bits to encode the tree structure. In order to encode the labels on the K leaves, we can use K bits. In sum, we need $3K-1$ bits to encode decision tree classifiers.

Then the complexity penalized ERM of decision trees is

$$\hat{f}_{CRM} = \min_{f\in\mathcal{F}}\{\widehat{R}(f) + \sqrt{\frac{3K-1+\log\frac{1}{\delta}}{2n}}\} = \min_{K}\{\min_{f\in\mathcal{F}_K}\widehat{R}(f) + \sqrt{\frac{3K-1+\log\frac{1}{\delta}}{2n}}\} \tag{11.18}$$

The error bound is

$$E[R(\hat{f}_{CRM})] \quad \leq \quad \min_{K}\{\min_{f\in\mathcal{F}_K} R(f) + \sqrt{\frac{3K-1+\log\frac{1}{\delta}}{2n}}\} + \delta \tag{11.19}$$

Thus, the complexity penalized procedure also performs model selection (pick the best K) for decision tree classifiers automatically.

11.5 Comparison of Histogram and Decision Trees Classifiers

In order to get the same "resolution", i.e. approximating error, for a $d-1$ dimensional decision boundary, histogram needes $m = K^2$ bins. Here we give a brief proof.

Let assume the true decision boundary is a function of $d-1$ out of the d coordinates and $b(x)$ is Lipschtiz where $x \in \mathbb{R}^{d-1}$. It means $|b(x) - b(x'| \leq L\|x - x'\|$ for some constant $L > 0$. For simplicity, lets consider $d = 2$.

Then, for histogram, the most number of bins that the boundary can intersect is $L\sqrt{m}$ since there are $\sqrt{m}$ bins along any coordinate. Then,

$$\inf_{f\in\mathcal{F}_m} R(f) - R^* \leq L\sqrt{m}\frac{1}{m}O(1) = O(\frac{1}{\sqrt{m}}) \tag{11.20}$$

since the best histogram classifier with m bins can incur at most $O(1)$ error in those bins and the probability of a test point falling in one such bin is $1/m$.

For decision tree with same resolution finest leaves i.e. the smallest leaves have size same as a histogram bin with m bins, the most number of finest resolution leaves that can intersect the boundary is $L\sqrt{m}$. Now there

are other leaves that may not intersect the boundary, but it can be shown that the total number of leaves of such a decision tree $K \leq 8L\sqrt{m}$. (A crude way to see a slightly looser bound is to realize that for every leaf that intersects the boundary, there are 4 leaves at each level that may need to be present in the tree, giving us $4L\sqrt{m}\log m$ bound on K. This can be improved by realizing that some of the leaves intersecting the boundary need to be co-located and hence share sibling leaves at higher levels. We skip the details.) Therefore,

$$\inf_{f \in \mathcal{F}_K} R(f) - R^* = L\sqrt{m}\frac{1}{m}O(1) = O(\frac{1}{\sqrt{m}}) = O(\frac{1}{K}) \tag{11.21}$$

So for histogram, the excess risk bound scales as

$$\min_m \frac{1}{\sqrt{m}} + \sqrt{\frac{m + \log\frac{1}{\delta}}{2n}} \asymp n^{-\frac{1}{4}}$$

where the optimum number of bins $m \asymp \sqrt{n}$.

For decision trees, the excess risk bound scales as

$$\min_K \frac{1}{K} + \sqrt{\frac{K + \log\frac{1}{\delta}}{2n}} \asymp n^{-\frac{1}{3}}$$

where the optimum number of leaves $K \asymp n^{1/3}$.

Thus, decision tree classifiers have error that converges faster with number of samples than histogram classifiers for well-behaved (d-1 dimensional) boundaries, which is usually the case. However, if the decision boundary is very complicated, essentially passing through all histogram bins, then the decision tree classifier with same approximation properties will need $K \asymp m$. As is intuitive, in this case, the decision tree classifiers are no better than histogram classifiers.