

Error Bounds in Countably Infinite Spaces

Lecturer: Rob Nowak

Scribe: Jarvis Haupt

1 Introduction

In the last lecture, we studied PAC bounds of the form $\forall f \in \mathcal{F}$, $|\mathcal{F}| < \infty$, and $\forall \delta > 0$ with probability at least $1 - \delta$

$$R(f) \leq \widehat{R}_n(f) + \sqrt{\frac{\log |\mathcal{F}| + \log(\frac{1}{\delta})}{2n}}$$

which led to error bounds of the form

$$E[R(\widehat{f}_n)] - \min_{f \in \mathcal{F}} R(f) \leq \sqrt{\frac{\log |\mathcal{F}| + \log(n) + 2}{4n}}$$

The key assumptions made in deriving the error bounds were:

- (i) bounded loss function
- (ii) finite collection of candidate functions

The bounds are valid for every P_{XY} and are called **distribution-free**.

2 Deriving Bounds for Countably Infinite Spaces

In this lecture we will generalize the previous results in a powerful way by developing bounds applicable to possibly infinite collections of candidates. To start let us suppose that \mathcal{F} is a countable, possibly infinite, collection of candidate functions. Assign a positive number $c(f)$ to each $f \in \mathcal{F}$, such that

$$\sum_{f \in \mathcal{F}} e^{-c(f)} < \infty$$

The numbers $c(f)$ can be interpreted as

- (i) measures of complexity
- (ii) $-\log$ of prior probabilities
- (iii) codelengths

In particular, if $P(f)$ is the prior probability of f then

$$e^{-(-\log p(f))} = p(f)$$

so $c(f) \equiv -\log p(f)$ produces

$$\sum_{f \in \mathcal{F}} e^{-c(f)} = \sum_{f \in \mathcal{F}} p(f) = 1$$

Now recall Hoeffding's inequality. For each f and every $\epsilon > 0$

$$P\left(R(f) - \widehat{R}_n(f) \geq \epsilon\right) \leq e^{-2n\epsilon^2}$$

or for every $\delta > 0$

$$P \left(R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log \left(\frac{1}{\delta} \right)}{2n}} \right) \leq \delta$$

Suppose $\delta > 0$ is specified. Using the values $c(f)$ for $f \in \mathcal{F}$, define

$$\delta(f) = e^{-c(f)} \delta$$

Then we have

$$P \left(R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log \left(\frac{1}{\delta(f)} \right)}{2n}} \right) \leq \delta(f)$$

Furthermore we can apply the union bound as follows

$$\begin{aligned} P \left(\bigcup_{f \in \mathcal{F}} R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log \left(\frac{1}{\delta(f)} \right)}{2n}} \right) &\leq \sum_{f \in \mathcal{F}} P \left(R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log \left(\frac{1}{\delta(f)} \right)}{2n}} \right) \\ &\leq \sum_{f \in \mathcal{F}} \delta(f) = \sum_{f \in \mathcal{F}} e^{-c(f)} \delta = \delta \end{aligned}$$

So we have that $\forall f \in \mathcal{F}$ and $\forall \delta > 0$ with probability at least $1 - \delta$

$$\begin{aligned} R(f) &\leq \hat{R}_n(f) + \sqrt{\frac{\log \left(\frac{1}{\delta(f)} \right)}{2n}} \\ &= \hat{R}_n(f) + \sqrt{\frac{c(f) + \log \left(\frac{1}{\delta(f)} \right)}{2n}} \end{aligned}$$

Special Case

Suppose \mathcal{F} is finite and $c(f) = \log |\mathcal{F}| \quad \forall f \in \mathcal{F}$. Then

$$\sum_{f \in \mathcal{F}} e^{-c(f)} = \sum_{f \in \mathcal{F}} e^{-\log |\mathcal{F}|} = \sum_{f \in \mathcal{F}} \frac{1}{|\mathcal{F}|} = 1$$

and

$$\delta(f) = \frac{\delta}{|\mathcal{F}|}$$

which implies $\forall f \in \mathcal{F}$, $|\mathcal{F}| < \infty$, and $\forall \delta > 0$ with probability at least $1 - \delta$

$$R(f) \leq \hat{R}_n(f) + \sqrt{\frac{\log |\mathcal{F}| + \log \left(\frac{1}{\delta(f)} \right)}{2n}}$$

Note that this is precisely the PAC bound we derived in the last lectures.

Choosing $c(f)$

The generalized bounds allow us to handle countably infinite collections of candidate functions, but we require that

$$\sum_{f \in \mathcal{F}} e^{-c(f)} < \infty$$

Of course, if $c(f) = -\log p(f)$ where $p(f)$ is a proper prior probability distribution then we have

$$\sum_{f \in \mathcal{F}} e^{-c(f)} = 1$$

However, it may be difficult to design a probability distribution over an infinite class of candidates. The coding perspective provides a very practical means to this end.

Assume that we have assigned a uniquely decodable binary code to each $f \in \mathcal{F}$, and let $c(f)$ denote the codelength for f . That is, the code for f is $c(f)$ bits long. A very useful class of uniquely decodable codes are called **prefix codes**.

Definition 1 A code is called a prefix code if no codeword is a prefix of any other codeword.

Example 1 (From Cover & Thomas '91)

Consider an alphabet of symbols, say A, B, C , and D and the codebooks below

Symbol	Singular Codebook	Nonsingular But Not Uniquely Decodable	Uniquely Decodable But Not a Prefix Code	Prefix Code
A	0	0	10	0
B	0	010	00	10
C	0	01	11	110
D	0	10	110	1110

In the singular codebook we assign the same codeword to each symbol - a system that is obviously flawed! In the second case, the codes are not singular but the codeword 010 could represent B or CA or AD. Hence it is not a uniquely decodable codebook.

The third and fourth cases are both examples of uniquely decodable codebooks, but the fourth has the added feature that no codeword is a prefix of another. Prefix codes can be decoded from left to right since each codeword is "self-punctuating" - in this case with a zero to indicate the end of each word.

To design a uniquely decodable codebook in general is as challenging as the problem of selecting $c(f)$ to satisfy

$$\sum_{f \in \mathcal{F}} e^{-c(f)} < \infty$$

However, prefix codes can often be easily designed or specified and they are inherently decodable. Moreover, prefix codes satisfy an important inequality called the **Kraft Inequality**.

3 The Kraft Inequality

For any binary prefix code, the codeword lengths c_1, c_2, \dots satisfy

$$\sum_{i=1}^{\infty} 2^{-c_i} \leq 1$$

Conversely, given any c_1, c_2, \dots satisfying the inequality above we can construct a prefix code with these codeword lengths. We will prove this result a bit later, but now let's see how this is useful in our learning problem.

Assume that we have assigned a binary prefix codeword to each $f \in \mathcal{F}$, and let $c(f)$ denote the bit-length of the codeword for f . Set $\delta(f) = 2^{-c(f)}\delta$. Then

$$\begin{aligned} P\left(\bigcup_{f \in \mathcal{F}} R(f) - \widehat{R}_n(f) \geq \sqrt{\frac{\log\left(\frac{1}{\delta(f)}\right)}{2n}}\right) &\leq \sum_{f \in \mathcal{F}} P\left(R(f) - \widehat{R}_n(f) \geq \sqrt{\frac{\log\left(\frac{1}{\delta(f)}\right)}{2n}}\right) \\ &\leq \sum_{f \in \mathcal{F}} \delta(f) = \sum_{f \in \mathcal{F}} 2^{-c(f)}\delta = \delta \end{aligned}$$

This implies that $\forall f \in \mathcal{F}$ and $\forall \delta > 0$ with probability at least $1 - \delta$

$$\begin{aligned} R(f) &\leq \widehat{R}_n(f) + \sqrt{\frac{\log\left(\frac{1}{\delta(f)}\right)}{2n}} \\ &= \widehat{R}_n(f) + \sqrt{\frac{c(f)\log 2 + \log\left(\frac{1}{\delta(f)}\right)}{2n}} \end{aligned}$$

Application

Let $\mathcal{F}_1, \mathcal{F}_2, \dots$ be a sequence of finite sets of candidate functions with $|\mathcal{F}_1| < |\mathcal{F}_2| < \dots$. We can design prefix codes as follows. Use the codes 0, 10, 110, 1110, ... to encode the subscript i in $|\mathcal{F}_i|$. For each class $|\mathcal{F}_i|$, construct a set of binary codewords of length $\lceil \log_2 |\mathcal{F}_i| \rceil$ to uniquely encode each function in \mathcal{F}_i . Then, encode any given function f by first using the code for i corresponding to the smallest \mathcal{F}_i that f belongs to, followed by the length $\lceil \log_2 |\mathcal{F}_i| \rceil$ codeword for $f \in \mathcal{F}_i$. This is a prefix code.

Example 2 *Histogram Classifiers*

$X=[0,1]^d$, $Y=\{0,1\}$. Let \mathcal{F}_k , $k=1, 2, \dots$ denote the collection of histogram classification rules with k equal volume bins. We can use the following codebook for the index k

k	Prefix Code
1	0
2	10
3	110
4	1110
.	.
.	.
.	.

And follow this codeword with $k=\log_2 |\mathcal{F}_k|$ bits to indicate which of the 2^k possible histogram rules is under consideration. Thus for any $f \in \mathcal{F}_k$ for some $k \geq 1$ there is a prefix code of length

$$c(f) = k + k = 2k \text{ bits}$$

It follows that $\forall f \in \bigcup_{k \geq 1} \mathcal{F}_k$ and $\forall \delta > 0$, with probability at least $1 - \delta$

$$R(f) \leq \widehat{R}_n(f) + \sqrt{\frac{2k_f \log 2 + \log\left(\frac{1}{\delta}\right)}{2n}}$$

where k_f is the number of bins in histogram corresponding to f . Contrast with the bound we had for the class of m bin histograms alone:

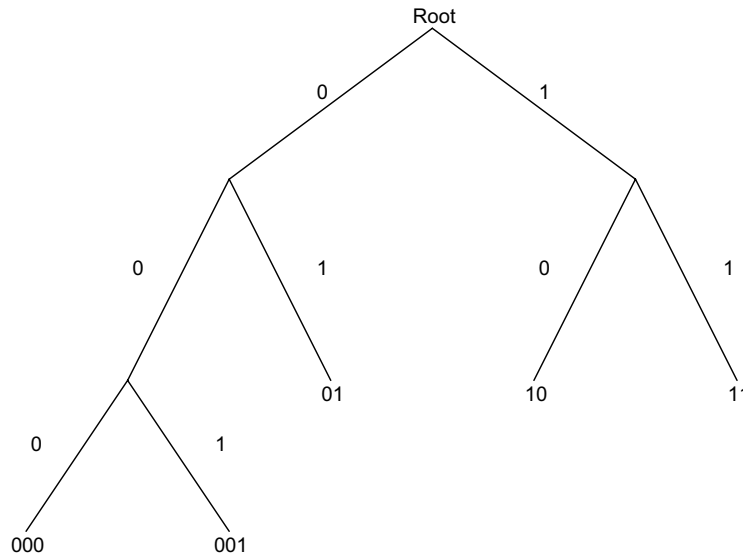
$\forall f \in \mathcal{F}_m$ and $\forall \delta > 0$, with probability $\geq 1 - \delta$

$$R(f) \leq \widehat{R}_n(f) + \sqrt{\frac{m \log 2 + \log\left(\frac{1}{\delta(f)}\right)}{2n}}$$

Notice the bound for all histograms rules is almost as good as the bound for on only the m -bin rules. That is, when $k_f = m$ the bounds are within a factor of $\sqrt{2}$. On the other hand, the new bound is a big improvement, since it also gives us a guide for selecting the number of bins.

Proof of the Kraft Inequality

We will prove that for any binary prefix code, the codeword lengths c_1, c_2, \dots satisfy $\sum_{k \geq 1} 2^{-c_k} \leq 1$. The converse is easy to prove also, but not central to our purposes here (for a proof, see Cover & Thomas '91). Consider a binary tree like the one shown below.



The sequence of bit values leading from the root to a leaf of the tree represents a codeword. The prefix condition implies that no codeword is a descendant of any other codeword in the tree. Let c_{max} be the length of the longest codeword (also the number of branches to the deepest leaf) in the tree.

Consider a leaf in the tree at level c_i . This node has $2^{c_{max}-c_i}$ possible descendants at level c_{max} . For each leaf, the set of descendants at level c_{max} is disjoint (since no codeword can be a prefix of another). Therefore, summing over all leaves

$$\sum_{c \in \text{leaves}} 2^{c_{max}-c_i} \leq 2^{c_{max}} \Rightarrow \sum_{c \in \text{leaves}} 2^{-c_i} \leq 1$$

Which proves the case when the number of codewords is finite.

Suppose now that we have a countably infinite number of codewords. Let $b_1 b_2 \dots b_{c_i}$ be the i^{th} codeword and let

$$r_i = \sum_{j=i}^{c_i} b_j 2^{-j}$$

be the real number corresponding to the binary expansion of the codeword. We can associate the interval $[r_i, r_i + 2^{-c_i})$ with the i^{th} codeword. This is the set of all real numbers whose binary expansion begins with $b_1 b_2 \dots b_{c_i}$. Since this is a subinterval of $[0, 1]$, and all such subintervals corresponding to prefix codewords are disjoint, the sum of their lengths must be less than or equal to 1. This proves the case where the number of codewords is infinite.