13. Structural Risk Minimization and the Kraft Inequality

ECE 830 & CS 761, Spring 2016

Countably Infinite Sets of Classifiers

Suppose that $\mathcal{F}$ is a countable, possibly infinite, collection of candidate functions.

Example: Histogram classifiers

$$\mathcal{F} = \bigcup_{k \geq 1} \mathcal{F}_k, \qquad \mathcal{F}_k \text{ is the set of histogram classifiers with } k \text{ bins}$$

Further suppose that we have some prior distribution p over this set, so that

$$p(f) \geq 0 \forall f \in \mathcal{F} \qquad \text{and} \qquad \sum_{f \in \mathcal{F}} p(f) = 1.$$

This provides two advantages:

1. By choosing $p(f)$ larger for certain f, we can preferentially treat those candidates
2. We do not need $\mathcal{F}$ to be finite and we only require $\sum_{f \in \mathcal{F}} p(f) = 1$

Let

$$c(f) = -\log p(f);$$

then we have

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

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

- -log of prior probabilities
- codelengths
- measures of complexity

PAC bound

Let $\delta(f) = \delta e^{-c(f)}$. We have that $\forall f \in \mathcal{F}$ and $\forall \delta > 0$ with probability at least 1-δ

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

Theorem: Complexity Regularized Model Selection

Let $\mathcal{F}$ be a collection of functions, and assign a positive number $c(f)$ to each $f \in \mathcal{F}$ such that $\sum_{f \in \mathcal{F}} e^{-c(f)} \leq 1$. Define the structural risk minimizer

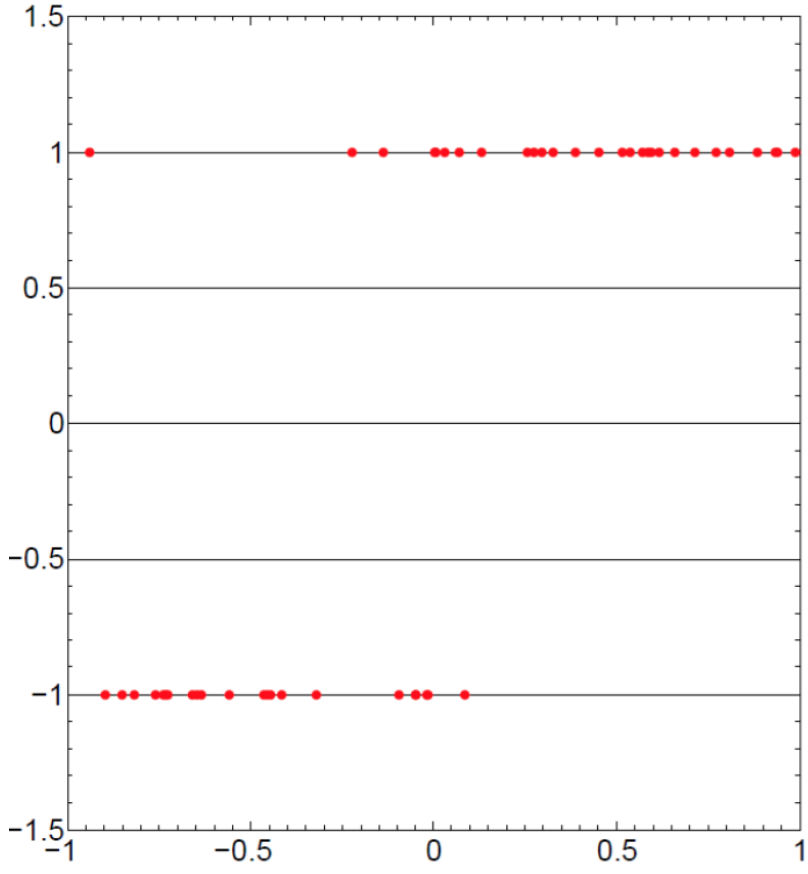
$$\widehat{f}_n = \arg\min_{f \in \mathcal{F}} \left\{ \widehat{R}_n(f) + \sqrt{\frac{c(f) + \frac{1}{2}\log n}{2n}} \right\}$$

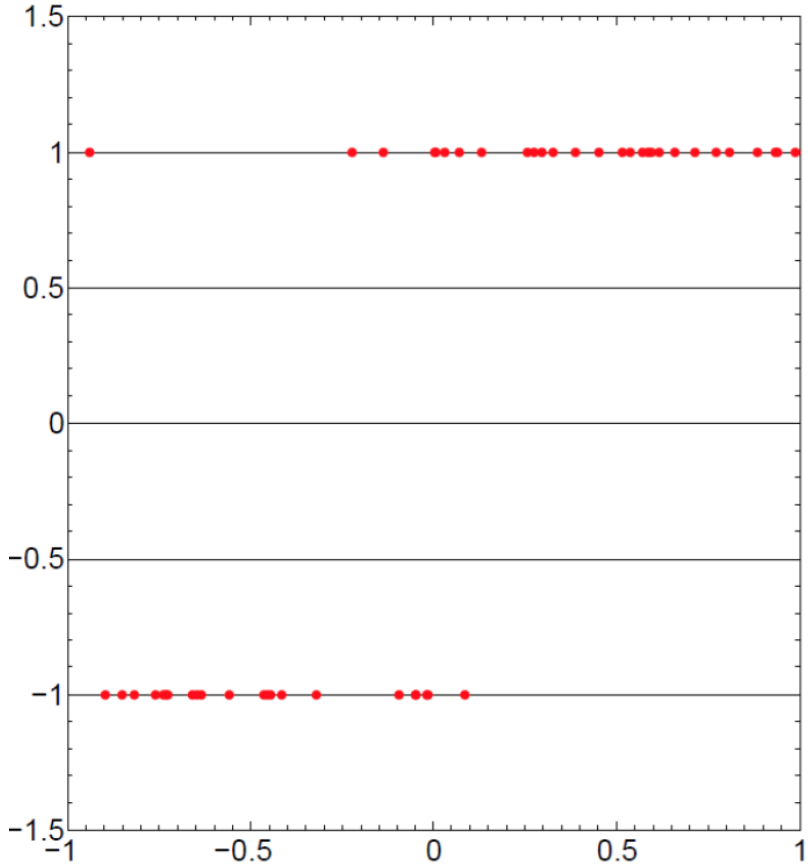
Then,

$$E[R(\widehat{f}_n)] \leq \inf_{f \in \mathcal{F}} \left\{ R(f) + \sqrt{\frac{c(f) + \frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}} \right\}$$

Key points

- For infinite sets of classifiers with infinite VC dimension, we previously had no bounds.
  - Theoretically, we can construct settings where the empirical risk minimizer would be arbitrarily bad.
  - Practically, if we have a large and complex set of predictors, ERM can over-fit the data and not generalize well.
- By using priors $p(f)$ or complexity measures $c(f)$, we can derive PAC bounds that were not possible before – BUT ERMs which do not use priors can still generalize poorly.


1.5
1
0.5
0
−0.5
−1
−1.5
−1
−0.5
0
0.5
1


1.5
1
0.5
0
−0.5
−1
−1.5
−1
−0.5
0
0.5
1

Key points (2)

- Using the PAC bounds, we can derive a useful alternative to ERM, called Structural Risk Minimization (SRM). SRM chooses a predictor f by balancing between
  - **(a)** the empirical risk
  - **(b)** the complexity of the predictor
- With collections of histogram classifiers, SRM automatically achieves nearly the same risk bound as ERM if ERM knew the best histogram resolution a priori.
- Aside from the criterion that $\sum_{f \in \mathcal{F}} e^{-c(f)} \leq 1$, we have not talked about how to choose $c(f)$.

Choosing the values $c(f)$

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.

In this lecture, we will consider choosing $c(f)$ according to how many bits are required to **encode** a predictor f.

The coding perspective provides a very practical means to this end.

Prefix codes

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: Prefix code

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

Example: Finite alphabet (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.

The Kraft Inequality

In general, designing 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**.

Kraft inequality

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

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

Conversely, given any c_1, c_2, ... satisfying the inequality above we can construct a prefix code with these codeword lengths.

Using the Kraft inequality

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

$$\mathbb{P}\left(\bigcup_{f\in\mathcal{F}} R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log(2/\delta(f))}{2n}}\right)$$

$$\leq \sum_{f\in\mathcal{F}} \mathbb{P}\left(R(f) - \hat{R}_n(f) \geq \sqrt{\frac{\log(2/\delta(f))}{2n}}\right) \leq \sum_{f\in\mathcal{F}} \delta(f) = \sum_{f\in\mathcal{F}} 2^{-c(f)}\delta = \delta$$

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

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

Example: Prefix codes for sequences of finite sets of classifiers

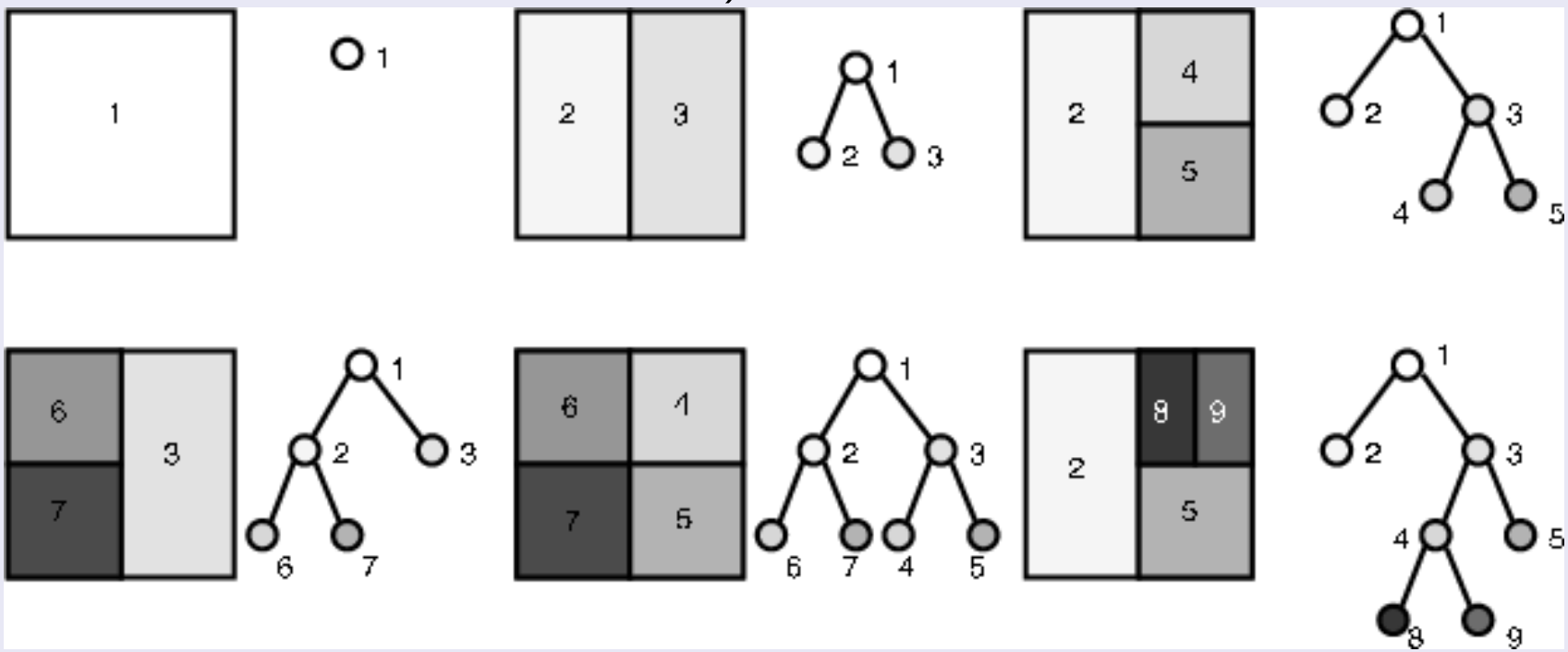
Let $\mathcal{F}_1$, $\mathcal{F}_2$, ... be a sequence of finite sets of candidate functions with $|\mathcal{F}_1| < |\mathcal{F}_2| < ...$ We can design prefix codes for $\mathcal{F} = \bigcup_{k \geq 1} \mathcal{F}_k$ as follows.

1. Use the codes 0, 10, 110, 1110, ... to encode the subscript i in $|\mathcal{F}_i|$.
2. 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 $\mathcal{F}_i$.
3. 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 of the codeword for $f \in \mathcal{F}_i$, which is $\lceil \log_2 |\mathcal{F}| \rceil$. You can easily show that this is a prefix code.

For a collection of histogram classifiers with different numbers of bins, a prefix code for f corresponding to a histogram with k bins would require k bits to specify the number of bins plus k bits to specify labels for each bin. Thus $c(f) = 2k$ whenever f has k bins satisfies $\sum_{f \in \mathcal{F}} 2^{-c(f)} \leq 1$.

Example: Prefix code for trees

Let $\mathcal{X} = [0,1]^d$, $\mathcal{Y} = \{0,1\}$, and let $\mathcal{F}$ denote the space of all binary dyadic decision trees. (Dyadic means that every cell is split in half when the tree branches.)



Example: Trees continued

How can we design a prefix code for the space of such trees?

Let k denote the number of leaves in the tree.

- First, we need to encode the structure of the tree. This can be accomplished using a pre-order traversal of the tree, assigning 0s to internal nodes and 1s to leaf nodes. The number of bits required is $2k - 1$.
- Second, for the internal nodes we need to encode the dimension along which the split occurs. There are d possible dimensions, so this requires $\log_2 d$ bits for each of the $k - 1$ internal nodes.
- Finally, for the leaf nodes we need to encode the label assigned to the corresponding cell, which requires one bit for each of the k leaf nodes.

The total number of bits in the prefix code is thus

$$2k - 1 + (k - 1) \log_2 d + k = O(k \log d).$$

Example: Polynomial fitting

Let $X = [0, 1]$, $Y = \mathbb{R}$, and let

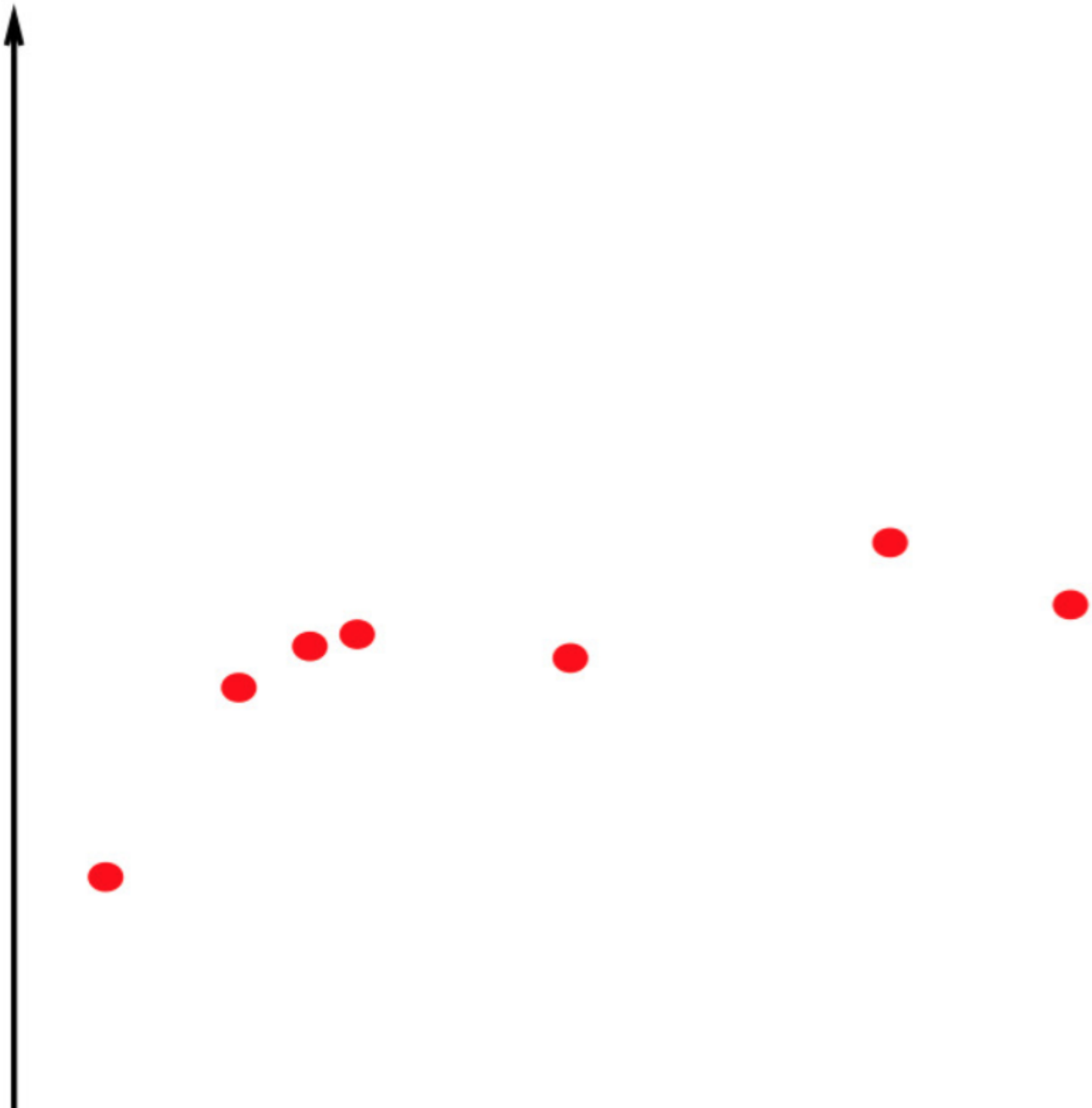
$$\mathcal{F}_{d,0} := \{f_w : \mathcal{X} \mapsto \mathcal{Y} : f_w(x) = w_1 + w_2 x + w_3 x^2 + \ldots + w_d x^{d-1}, w \in [-1, 1]\}$$

be a set of order-d polynomials. This is an uncountably infinite set. We will discretize by only allowing w to take on values in a finite set. Specifically, each of the d elements of w will be represented with q bits, so there can be 2^{dq} different w. Let $\mathcal{W}$ denote the set of these quantized w, and define

$$\mathcal{F}_d := \{f_w : \mathcal{X} \mapsto \mathcal{Y} : f_w(x) = w_1 + w_2 x + w_3 x^2 + \ldots + w_d x^{d-1}, w \in \mathcal{W}\}$$

Note $|\mathcal{F}_d| = 2^{dq}$, so if we use ERM we find that the expected risk is bounded by

$$\begin{aligned}\mathbb{E}R(\hat{f}_{ERM}) &\leq \min_{f \in \mathcal{F}} R(f) + O\left(\sqrt{\frac{\log |\mathcal{F}_d| + \log n}{n}}\right) \\ &= \min_{f \in \mathcal{F}} R(f) + O\left(\sqrt{\frac{dq \log n}{n}}\right).\end{aligned}$$



Example: Polynomial fitting continued

Alternatively, consider structural risk minimization over the much larger set $\mathcal{F} = \bigcup_{k \geq 0} \mathcal{F}_k$ where we do not fix the polynomial order ahead of time. To design a prefix code, we need to encode (a) the polynomial order k and (b) the k polynomial coefficient values. This leads to

$$c(f_w) = k + kq.$$

So if we use SRM we find that the expected risk is bounded by

$$\mathbb{E}R(\hat{f}_{SRM}) \leq \min_{k \geq 0} \left\{ \min_{\substack{w \in \mathcal{W}, \\ \text{order}-k}} \left[R(f_w) + O\left(\sqrt{\frac{c(f_w) \log 2 + \log n}{n}} \right) \right] \right\}$$

Example: Polynomial fitting continued

If we imagine that for some $k^* < d$ that there is some order-k^* polynomial with coefficients in $\mathcal{W}$ so that $R(f_w) = 0$, then

$$\mathbb{E}R(\hat{f}_{ERM}) \leq O\left(\sqrt{\frac{dq \log n}{n}}\right)$$

$$\mathbb{E}R(\hat{f}_{SRM}) \leq O\left(\sqrt{\frac{k^*(q+1) \log n}{n}}\right)$$

Example: Logistic regression

Let $\mathcal{X} = [0,1]^d$, $\mathcal{Y} = \{0,1\}$, and let

$$\mathcal{F}_0 := \{f_w : \mathcal{X} \mapsto \mathcal{Y} : f_w(x) = \mathbf{1}_{\{\frac{1}{1+e^{\langle x,w\rangle}} > 1/2\}}, w \in [-1,1]^d\}$$

This is an uncountably infinite set. We will discretize by only allowing w to take on values in a finite set. Specifically, each of the d elements of w will be represented with q bits, so there can be 2^{dq} different w. Let $\mathcal{W}$ denote the set of these quantized w, and define

$$\mathcal{F} := \{f_w : \mathcal{X} \mapsto \mathcal{Y} : f_w(x) = \mathbf{1}_{\{(1+e^{\langle x,w\rangle})^{-1} > 1/2\}}, w \in \mathcal{W}\}$$

Note $|\mathcal{F}| = 2^{dq}$, so if we use ERM we find that the expected risk is bounded by

$$\begin{aligned}\mathbb{E}R(\hat{f}_{ERM}) &\leq \min_{f\in\mathcal{F}} R(f) + O\left(\sqrt{\frac{\log|\mathcal{F}| + \log n}{n}}\right) \\ &= \min_{f\in\mathcal{F}} R(f) + O\left(\sqrt{\frac{dq\log n}{n}}\right).\end{aligned}$$

Example: Logistic regression continued

Alternatively, consider structural risk minimization. Note that some $w \in \mathcal{W}$ will have some zero-valued elements; we will develop a code which is shorter for sparser w (with more zeros). Let k denote the number of non-zeros in w. To design a prefix code, we need to encode (a) the value of k, (b) the locations of the non-zeros and (c) their values. This leads to

$$c(f_w) = k + k \log d + kq \text{ for } k\text{-sparse } w.$$

So if we use SRM we find that the expected risk is bounded by

$$\mathbb{E}R(\hat{f}_{SRM}) \leq \min_{k \geq 0} \left\{ \min_{\substack{w \in \mathcal{W}, \\ k-\text{sparse}}} \left[R(f_w) + O\left(\sqrt{\frac{c(f_w) \log 2 + \log n}{n}} \right) \right] \right\}$$

Example: Logistic regression continued

If we imagine that for some $k^* < d$ that there is some k^*-sparse $w \in \mathcal{W}$ so that $R(f_w) = 0$, then

$$\mathbb{E}R(\hat{f}_{ERM}) \leq O\left(\sqrt{\frac{dq \log n}{n}}\right)$$

$$\mathbb{E}R(\hat{f}_{SRM}) \leq O\left(\sqrt{\frac{k^*(q + \log d) \log n}{n}}\right)$$