10-704: Information Processing and Learning **Fall 2016**

Lecture 12: Oct 10

Lecturer: Aarti Singh

Note: *These notes are based on scribed notes from Spring15 offering of this course. LaTeX template courtesy of UC Berkeley EECS dept.*

12.1 Review

Complexity Penalized ERM aka Structural Risk Minimization via prefix codes

Last time, we talked about Complexity Penalized ERM via prefix codes for loss functions that are bounded. This includes classification under 0-1 loss. For bounded loss functions, the complexity penalized ERM predictor is given as follows:

$$\widehat{f} = \arg\min_{f \in \mathcal{F}} \left\{ \widehat{R}(f) + \sqrt{\frac{c(f) + \ln(2/\delta)}{2n}} \right\}$$

where $\widehat{R}(f) = \frac{1}{n}\sum_{i=1}^{n} \text{loss}(f(X_i), Y_i))$ for n i.i.d. samples $\{X_i, Y_i\}_{i=1}^n$, $c(f)$ is the prefix codelength for encoding the predictor f and $\delta \in (0,1)$ is a confidence parameter. For binary classification, for example, the 0/1 loss is $\text{loss}(f(X_i), Y_i) = 1_{f(X_i) \neq Y_i}$. We also showed the following bound on the expected excess risk for the complexity penalized ERM predictor:

$$\mathbb{E}[R(\widehat{f})] - R^* \leq \min_{f \in \mathcal{F}} \left\{ R(f) - R^* + \sqrt{\frac{c(f) + \ln(1/\delta)}{2n}} \right\} + \delta$$

for all $\delta \in (0,1)$ where R^* is the Bayes optimal risk (i.e. risk of the best possible predictor not restricted to be in $\mathcal{F}$). The first two terms can be regarded as approximation error (bias) and the third term as estimation error (variance). Thus, complexity penalized ERM automatically does model selection, i.e. picks a predictor with performance comparable to the one in the class $\mathcal{F}$ that balances approximation and estimation errors. On the other hand, simply doing ERM is prone to over-fitting, i.e. it will pick an overly complex predictor with small empirical risk but large codelength, resulting in small approximation error (since a complex predictor can better approximate a function not in the class) but large estimation error.

Application to Classification

We also studied two examples of complexity penalized ERM via prefix codes - histogram and dyadic decision tree classifiers. We derived prefix codes for them and studied approximation properties for Lipschitz class of boundaries, demonstrating rates of convergence which indicate that decision trees can outperform histogram classifiers when the boundary is well-behaved.

12.2 Complexity Penalized ERM for Regression and Density Estimation

To derive performance bounds on complexity penalized ERM predictors under bounded losses, we had used Hoeffding's concentration inequality to control the deviation of true and empirical risk. For unbounded loss functions, we can use other concentration inequalities such as Craig-Bernstein inequality which under some moment conditions on the variables imply a tighter control on the deviation of true and empirical risk: With probability $\geq 1-\delta$, for all $f \in \mathcal{F}$

$$R(f) \leq \widehat{R}(f) + \frac{c(f) + \ln(1\delta)}{n}$$

Notice the absence of square-root on the deviation bound. The complexity penalized ERM aka Structural Risk Minimization (SRM) rule is given as

$$\widehat{f} = \arg\min_{f \in \mathcal{F}} \left\{ \widehat{R}(f) + \frac{c(f) + \ln(1/\delta)}{n} \right\}$$

and its expected excess risk can be bounded as follows:

$$\mathbb{E}[R(\widehat{f})] - R^* \leq \min_{f \in \mathcal{F}} \left\{ R(f) - R^* + \frac{c(f) + \ln(1/\delta)}{n} \right\} + \delta$$

For a derivation, see for example http://nowak.ece.wisc.edu/SLT09/lecture12.pdf.

Examples of unbounded loss functions arise in Regression/de-noising where the squared loss $\text{loss}(f(X_i), Y_i) = (f(X_i) - Y_i)^2$ is used, and density estimation where negative log likelihood loss $\text{loss}(f(X_i)) = -\log f(X_i)$ is used. Lets consider application of these bounds to histogram regression, wavelet denoising and density estimation under a Markov chain model.

All these are examples of (uncountable) infinite classes of models since the parameters in these models can take on any real value (for example, in histogram regression, instead of a binary label, a model assigns a real value to each bin). To enable encoding of such models, we need to consider a finite or countably infinite subset of models $\bar{\mathcal{F}} \subset \mathcal{F}$ that cover the original class of models $\mathcal{F}$. One way to get such a class is quantization, where the real parameters are quantized to a discrete collection of values. Notice that we can still get an error bound (since $\bar{\mathcal{F}} \subset \mathcal{F}$ and hence minimization over $\bar{\mathcal{F}}$ upper bounds the error bound given above)

$$\mathbb{E}[R(\widehat{f})] - R^* \leq \min_{f \in \bar{\mathcal{F}}} \left\{ R(f) - R^* + \frac{c(f) + \ln(1/\delta)}{n} \right\} + \delta$$

that is usually not too worse if the quantization is fine enough and shrinking to 0 as $n \to \infty$.

12.2.1 Application to Histogram regression

Let $\mathcal{F}_m$ denote the class of histogram regressors with m bins. We will consider models where the values in each bin are bounded between $[-M, M]$. Let $\bar{\mathcal{F}}_m$ denote the class of histogram regressors where the value in each of the m bins is quantized to an accuracy of $2M/\Delta$. Hence, the number of possible values for each bin in a quantized model is Δ, resulting in total $|\bar{\mathcal{F}}_m| = \Delta^m$ quantized histogram regressors.

To encode Δ^m different classifiers, we need $\log \Delta^m = O(m)$ bits using uniform coding - notice that this is a prefix code. In addition, we need to encode the integer m denoting the histogram resolution, which needs $\log m$ bits. Appending these bits to the uniform code still leads to a prefix code. Therefore, we need $c(f) = O(m)$ bits to encode $f \in \bar{\mathcal{F}}_m$. [1] We will consider the countably infinite class $\bar{\mathcal{F}} = \cup_m \bar{\mathcal{F}}_m$.

[1] Though a better code can result in tighter bounds, such an encoding is rate optimal (gets best possible dependence of error on number of samples) and we will only care about right order of number of bits and resulting error.

Plugging this in the earlier expression, we have the complexity penalized ERM or SRM predictor.

$$\hat{f} = \arg\min_{f \in \bar{\mathcal{F}}} \left\{ \hat{R}(f) + \frac{O(m) + \log \frac{1}{\delta}}{n} \right\} = \arg\min_{m} \left\{ \min_{f \in \bar{\mathcal{F}}_m} \hat{R}(f) + \frac{O(m) + \log \frac{1}{\delta}}{n} \right\} \tag{12.1}$$

The error bound is

$$E[R(\hat{f})] - R^* \quad \leq \quad \min_{m} \{ \min_{f \in \bar{\mathcal{F}}_m} R(f) - R^* + \frac{O(m) + \log \frac{1}{\delta}}{n} \} + \delta \tag{12.2}$$

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

We can derive the rate of error convergence for functions that are Lipschitz i.e. $|f^*(x) - f^*(x')| \leq L\|x - x'\|$ [2]. To characterize the approximation error $\min_{f \in \bar{\mathcal{F}}_m} R(f) - R^*$ for this class, let $\bar{f} = \arg\min_{f \in \bar{\mathcal{F}}_m} R(f)$ and $f = \arg\min_{f \in \mathcal{F}_m} R(f)$. Also notice that $R(\bar{f}) - R^* = E[(\bar{f}(X) - f^*(X)^2)] \leq E[(|\bar{f}(X) - f(X)| + |f(X) - f^*(X)|)^2] \leq E[(M/\Delta + |f(X) - f^*(X)|)^2]$. The best unquantized histogram f of resolution m is a constant in each bin while f^* is Lipschitz, hence $|f(x) - f^*(x)| = O(1/m^{1/d})$ since the volume of a between any two points in a bin is $O(1/m^{1/d})$. Thus, we get that

$$R(\bar{f}) - R^* = O\left(\left(\frac{1}{\Delta} + \frac{1}{m^{1/d}} \right)^2 \right)$$

and

$$E[R(\hat{f})] - R^* = \min_{m} O\left(\left(\frac{1}{\Delta} + \frac{1}{m^{1/d}} \right)^2 + \frac{m}{n} \right) = O\left(1/\Delta^2 + n^{-2/(d+2)} \right)$$

since the optimal $m = \Theta(n^{d/(d+2)})$. In fact, $n^{-2/(d+2)}$ is the best possible rate of mean square error convergence for any estimator (we will see how to derive such lower bounds later in the course), and hence the histogram regressor is optimal assuming the quantization level Δ is set to satisfy $\Omega(n^{1/(d+2)})$.

12.2.2 Application to Wavelet de-noising

Another approach to perform regression is via basis expansion, e.g. the models can be given by a linear combination of fourier basis functions or wavelet basis functions. Fourier basis provides a good approximation for smooth functions. Wavelets basis is good for representing inhomogeneous functions such as piece-wise smooth functions or functions supported locally on a small part of the domain. For example, the Haar wavelet basis (shown below) provides a good approximation for piece-wise constant functions.

A wavelet basis consists of orthonormal basis functions indexed by a scale parameter $\ell \in \{0, 1, \dots\}$ and a location parameter $k \in \{1, 2, \dots, 2^\ell\}$. For example, there is only one location at scale $\ell = 0$ and a single basis element which is supported on the entire domain and for Haar basis is a positive value on one half of the domain and a negative value on the other half. The next scale corresponding to $\ell = 1$ has two basis elements at locations $k = 1, 2$ that consist of the first element shrunk by half and supported on first half (positive on first quarter and negative on second quarter for Haar basis) and second half of the domain (positive on third quarter and negative on fourth quarter for Haar basis). And so on.

We represent a function in the wavelet basis as follows:

$$f(x) = \sum_{\ell \geq 0} \sum_{k=1}^{2^\ell} a_{\ell k} \phi_{\ell k}(x) + c_0$$

[2]Notice that unlike classification where the boundary was Lipschitz, here we are modeling the regression function f^* as Lipschitz.

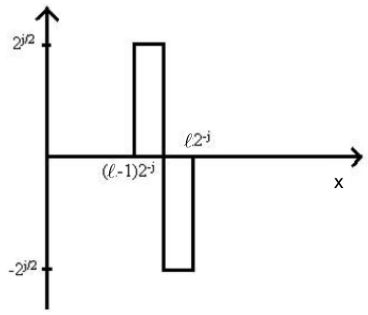

Figure 12.1: A haar wavelet basis element, $\phi_{\ell k}(x)$

where ϕ_{jk} is the wavelet basis element at scale ℓ and location k. Notice that the Haar wavelet basis is orthonormal, i.e. $\int \phi(x)\phi'(x)dx = 0$ unless $\phi(x) = \phi'(x)$ in which case $\int \phi^2(x)dx = 1$. Also, orthonormality implies that we have $a_{\ell k} = \int f(x)\phi_{\ell k}(x)dx =< f, \phi_{\ell k} >$.

Wavelet basis elements satisfy certain additional constraints called *vanishing moments*. For example, the Haar wavelet basis satisfies $\int \phi_{\ell k}(x)dx = 0$. More generally, a s^{th}-order wavelet basis satisfies $\int x^r \phi_{\ell k}(x)dx = 0$ for all $r \leq s$ and is said to possess s vanishing moments. A nice consequence of this property is that if the function f is smooth over the support of $\phi_{\ell k}$, say is it constant for haar basis or looks like x^r for higher-order basis, then the corresponding coefficient $a_{\ell k} = 0$. Thus, wavelet basis provides a *sparse representation* for functions that are inhomogeneous. For example, the only non-zero coefficients in the Haar wavelet basis for a piece-wise constant function are the ones whose basis elements are supported over discontinuities of the function.

Typically, the function f is only evaluated at n samples and in this case, only wavelet basis up to resolution $\ell = \log_2 n - 1$ suffice i.e. n basis elements and their corresponding coefficients.

Choosing a wavelet estimator for data $\{X_i, Y_i\}_{i=1}^n$ correspond to picking a function (model) f expressed in terms of its wavelet basis expansion. We can pick a good model from this class by using complexity penalized ERM:

$$\widehat{f} = \arg\min_{f \in \mathcal{F}} \{\widehat{R}(f) + \frac{c(f) + \ln(1/\delta)}{n}\}$$

where $\widehat{R}(f) = \frac{1}{n}\sum_{i=1}^n (Y_i - f(X_i))^2$ and $c(f)$ is the length of a prefix code for f.

How can we encode a wavelet estimator f? Notice that encoding f is equivalent to encoding the non-zero coefficients - their scale, location and the value. For each non-zero coefficient, there are n possible combinations of scale, location (since there are n samples) and hence that can be encoded simply with $\log_2 n$ bits. The value of each coefficient can be any real (but bounded) number and representing this perfectly could require infinite bits, but again since we have only n samples, it suffices to quantize the wavelet coefficients and encode them. Specifically, we consider the class $\overline{\mathcal{F}}$ where the functions f have wavelet representations with coefficients quantized to an accuracy of $2M/\sqrt{n}$ if the original coefficients are bounded between $[-M, M]$ where M is some constant. Then a coefficient can be encoded with $\log_2(\sqrt{n}) = \frac{1}{2}\log_2 n$ bits. By doing this, we lose a little bit (order $1/n$ in the approximation error, which is best possible accuracy one can hope for anyways) but gain a lot in the estimation error (since we can encode functions in $\overline{\mathcal{F}}$ using few bits). Therefore,

$$c(f) = \frac{3}{2}\log_2 n \times \text{number of non-zero coeffs.}$$

If we let $\mathbf{a}$ is a column vector containing the coefficients and c_0, then $c(f) = \frac{3}{2}\|\mathbf{a}\|_0 \log_2 n$. The estimated

coefficients are given as:

$$\widehat{\mathbf{a}} = \arg\min_{\mathbf{a}\in A} \{\|Y - \Phi(X)\mathbf{a}\|^2 + \frac{3}{2}\|\mathbf{a}\|_0 \log_2 n\}$$

where $A = [-M+M/\sqrt{n}, -M+3M/\sqrt{n}, ..., M-M/\sqrt{n}]$ is the set of quantized coefficients, $Y = [Y_1, Y_2, \ldots, Y_n]$, and $\Phi(X) = [\Phi(X_1); \Phi(X_2); \ldots; \Phi(X_n)]$ is an $n \times n$ matrix containing the n basis functions sampled at uniformly spaced points $X_1, \ldots, X_n$. It can be shown that the solution $\widehat{\mathbf{a}}$ can be written in closed form as a hard-thresholding estimate and hence is easy to compute, i.e. if $Y = \Phi(X)\mathbf{b}$, then

$$\widehat{a}_j = b_j \cdot \mathbf{1}\left(|b_j| \geq \sqrt{1.5 \log_2 n}\right)$$

where $\mathbf{1}(S) = 1$ if S is true and 0 otherwise. To see this, notice that $\|Y - \Phi(X)\mathbf{a}\|^2 = \|\Phi(X)\mathbf{b} - \Phi(X)\mathbf{a}\|^2 = \|\mathbf{b} - \mathbf{a}\|^2$ since orthonormality of basis functions implies $\Phi(X)^\top \Phi(X) = I$. Now, notice that the solution to $\arg\min_{\mathbf{a}\in A} \{\|\mathbf{b} - \mathbf{a}\|^2 + \frac{3}{2}\|\mathbf{a}\|_0 \log_2 n\}$ can only be $a_j = b_j$ (in which case error contribution is $\frac{3}{2}\log_2 n$) or $a_j = 0$ (in which case error contribution is b_i^2). Therefore, the optimal $\widehat{a}_j = b_j \cdot \mathbf{1}\left(|b_j| \geq \sqrt{\frac{3}{2}\log_2 n}\right)$.

Thus, a wavelet estimator is often referred to as a "non-linear" estimator since the basis elements used in the representation are the ones with largest coefficients and not simply the first few basis elements as is typical.

We can also bound the expected excess risk of the complexity penalized wavelet estimator.

$$\begin{aligned}
\mathbb{E}[R(\widehat{f})] - R^* &\leq \min_{f\in\mathcal{F}} \left\{ R(f) - R^* + \frac{c(f) + \ln(1/\delta)}{n} \right\} + \delta \\
&\leq \min_{f\in\overline{\mathcal{F}}} \left\{ R(f) - R^* + \frac{c(f) + \ln(1/\delta)}{n} \right\} + \delta
\end{aligned}$$

Second inequality follows since $\overline{\mathcal{F}} \subset \mathcal{F}$. If the true underlying function f^* has d discontinuities and we are using a Haar wavelet basis, then the estimation error behaves like $(\|\mathbf{a}\|_0 \log_2 n)/n = O((d \log_2^2 n)/n)$ since every discontinuity results in no more than $\log_2 n$ non-zero coefficients (all basis elements with supports that include the point of discontinuity). Notice that $R(f) - R^* = \mathbb{E}[(f(X) - f^*(X))^2]$ and hence it can be shown that the approximation error behaves like $O(d/n)$ since the best wavelet estimator is quantized to $1/\sqrt{n}$ in coefficient value and to $1/n$ spatially (i.e. in the d regions of discontinuity, each of size $1/n$, the error is $O(1)$, and elsewhere the squared error is $1/n$). Thus, the overall mean square error behaves like $O((d \log_2^2 n)/n)$, which is the parametric rate of error convergence $O(\text{no. of parameters}/n)$ up to log factors.