10-704 Homework 3
Due: Thursday 4/14/2015

Instructions: Turn in your homework in class on Thursday 4/14/2015

1. **Wavelets and Complexity Penalized ERM** In this problem we will analyze the rate of convergence of the wavelet estimator for denoising.

   The Haar wavelets are piecewise constant functions over $[0, 1)$ defined by:

   $$\psi_{j,k}(x) = 2^{j/2}\left(\mathbf{1}[x \in [2^{-j}(k-1), 2^{-j}(k-1/2)) - \mathbf{1}[x \in [2^{-j}(k-1/2), 2^{-j}k)\right)$$

   For $j \in \mathbb{N} \cup \{0\}$ and $1 \leq k \leq 2^j$. We also have the scaling function $\psi_{0,0}(x) = \mathbf{1}[x \in [0,1)]$.

   We saw in class that these functions form an orthonormal basis for $[0, 1)$, we can therefore write any function supported on $[0, 1)$ as:

   $$f(x) = \sum_{j \geq 0} \sum_{k=1}^{2^j} a_{j,k}\psi_{j,k}(x) + a_{0,0}\phi_{0,0}(x)$$

   For any function f let $\Pi_l f$ denote the projection of f onto the top l scales of the wavelet basis. That is:

   $$(\Pi_l f)(x) = \sum_{j=0}^{l} \sum_{k=1}^{2^j} \langle \psi_{j,k}, f\rangle \psi_{j,k}(x) + \langle \psi_{0,0}, f\rangle \psi_{0,0}(x)$$

   Typically we have to quantize the wavelet coefficients to some precision, so now let $\Pi_{l,\epsilon}$ denote the projection onto the top l scales but where all coefficients are quantized.

   $$\Pi_{l,\epsilon} f = \min_{a_{j,k} = \beta_{j,k}\epsilon,\ \beta_{j,k} \in \mathbb{Z}} \|f - \sum_{j=0}^{l} \sum_{k=1}^{2^j} a_{j,k}\psi_{j,k} - a_{0,0}\psi_{0,0}\|_2$$

   where $\mathbb{Z}$ denotes the set of integers. In words, $\Pi_{l,\epsilon} f$ is the best approximation (in the L_2 sense) to f over all Haar wavelet representations with scale at most l and with coefficients that are multiples of ϵ.

   Let $\mathcal{F}_{s,M}$ be the set of piecewise constant functions supported over $[0, 1)$ with at most s discontinuities and with ℓ_∞ norm bounded by M (i.e. $\sup_x |f(x)| \leq M$ for all $f \in \mathcal{F}_{s,M}$). Consider an $f \in \mathcal{F}_{s,M}$.

   (a) How many non-zero wavelet coefficients does $\Pi_l f$ have? A good upper bound is sufficient.

(b) Give an upper bound on the approximation error $\|f - \Pi_l f\|_2^2$.

(c) How many non-zero wavelet coefficients does $\Pi_{l,\epsilon} f$ have? Again, a good upper bound suffices.

(d) Give an upper bound on the approximation error $\|f - \Pi_{l,\epsilon} f\|_2^2$.

(e) For a function $f \in \mathcal{F}_{s,M}$. How many bits $c(f)$ does it take to encode $\Pi_{l,\epsilon} f$?

(f) Suppose we are given noisy samples $Y_i = f^*(X_i) + \epsilon_i$ for $i = 1, \ldots, n$ where $f^* \in \mathcal{F}_{s,M}$ and $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ are i.i.d. We estimate the true function f^* using complexity penalized ERM (CRM) over the class of estimators $g \in \mathcal{G} := \{\Pi_l f : f \text{ supported on } [0,1)\}$. The CRM rule for wavelet denoising is

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

(Recall we discussed how this optimization can be implemented using a simple hard-thresholding procedure).

For the squared loss, $R(g) = \mathbb{E}[(g(X) - Y)^2]$ and $R(g) - R^* = \mathbb{E}[(g(X) - f^*(X))^2] = \|g - f^*\|_2^2$ assuming X_i are uniformly distributed on $[0,1)$. Using this, we have the following bound under squared loss for the CRM rule for wavelet denoising:

$$\begin{aligned} \|\widehat{g} - f^*\|_2^2 &\le \min_{g \in \mathcal{G}} \left\{ \|g - f^*\|_2^2 + \frac{c(g) + \ln(1/\delta)}{n} \right\} + \delta \\ &\le \min_{g \in \mathcal{G}_\epsilon} \left\{ \|g - f^*\|_2^2 + \frac{c(g) + \ln(1/\delta)}{n} \right\} + \delta \end{aligned}$$

for all $\delta > 0$ and where $\mathcal{G}_\epsilon = \{\Pi_{l,\epsilon} f : f \text{ supported on } [0,1)\}$. Use the previous results in (d), (e) to show that the rate of error convergence of the CRM rule for wavelet denoising is $O((s \log^2 n)/n)$, if we quantize to $\epsilon = 1/\sqrt{n}$ and $\ell = \log_2 n - 1$ as discussed in class. This is essentially a parametric rate where s serves as the number of parameters.

2. **Universal Prediction** Recall the setting for the exponential weights algorithm. We had a finite class of predictors $\mathcal{F}$, with predictors $f_1, \ldots, f_N$ and play a T round game, starting with distribution $q_1 = (1/N, \ldots, 1/N)$ and $\eta = \sqrt{8 \ln N / T}$. At round t, nature reveals expert advice x_t, we draw $i_t \sim q_t$ and play $\widehat{y}_t = f_{i_t}(x_t)$ and suffer loss $l(\widehat{y}_t, y_t)$ for some true label y_t. In the exponential weights algorithm, we updated the distribution q_t as:

$$q_{t+1}(i) \propto q_t(i) \times \exp\left\{-\eta l(f_i(x_t), y_t)\right\}$$

We can extend the algorithm to an infinite class of predictors using a prior π on $\mathcal{F}$ instead of the uniform prior. In this problem, you will demonstrate a bound on the expected regret of the algorithm, i.e. on

$$\frac{1}{T}\mathbb{E}\sum_{t=1}^{T} l(\widehat{y}_t, y_t) - \inf_{f \in \mathcal{F}} \frac{1}{T}\sum_{t=1}^{T} l(f(x_t), y_t),$$

for any loss l that is bounded in $[0,1]$. The expectation is over the randomness in our algorithm.

Let $W_0 = \sum_i \pi_i = 1$ and $W_t = \sum_i \pi_i e^{-\eta L_t^i}$ where $L_t^i = \sum_{\tau=1}^t l(f_i(x_\tau), y_\tau)$ is the cumulative loss of expert f_i.

(a) Show that $\ln W_T \geq -\min_i[\eta L_T^i + \log \pi_i]$.

(b) Also show that $W_t/W_{t-1} = \ln \mathbb{E}_{i_t \sim q_t} \exp(-\eta l(f_{i_t}(x_t), y_t))$.

(c) We will now use an inequality that bounds the moment generating function for a bounded random variable $X \in [a,b]$:

$$\ln \mathbb{E}e^{sX} \leq s\mathbb{E}X + \frac{s^2(b-a)^2}{8}$$

Note that this is the exact inequality that we use to prove Hoeffding's inequality. Since the losses $l \in [0,1]$, argue that this inequality implies

$$\ln W_T \leq -\eta \sum_{t=1}^T \mathbb{E}_{i_t} l(f_{i_t}(x_t), y_t) + \frac{\eta^2 T}{8}$$

(d) Combine the previous results (upper and lower bounds on $\ln W_T$) to argue that for any prior π on $\mathcal{F}$ we have for an appropriate setting of the parameter η:

$$\frac{1}{T}\mathbb{E}\sum_{t=1}^T l(\widehat{y}_t, y_t) \leq \inf_{f \in \mathcal{F}} \left\{ \frac{1}{T}\sum_{t=1}^T l(f(x_t), y_t) + \frac{1 + \log(1/\pi(f))}{\sqrt{8T}} \right\}$$

The bound shows that the performance of the algorithm is close to the best penalized performance of any predictor where the penalization reflects our prior belief in the expert. *Remark: A slightly better regret bound with dependence on* $\sqrt{\log(1/\pi(f))}$ *is also possible.*

3. **Arithmetic Coding**

Arithmetic Coding is a method to encode a (possibly very long) sequence of symbols from an alphabet. Recall that while the Huffman/Shannon codes have desirable code lenghts, they may be computationally inefficient for very long sequences. In addition, arithmetic coding also allows for sequential encoding and decoding.

In this question you will implement Arithmetic Coding. We have provided starter code in Matlab, but you may use any language of your choice. The starter code is downloadable from the Homework page. These are the details of the implementation.

- Our alphabet has 4 symbols $\mathcal{X} = \{2,3,4,5\}$.
- We will use the symbol "1" as a terminating symbol - i.e. the encoder terminates each sequence with "1" and the decoder, upon seeing a "1" stops decoding.

- The symbols $\mathcal{X} \cup \{1\}$ are drawn from a prior distribution p given in the starter code.
- You should write a function `arithmeticEncode` that takes in as arguments p and a sequence of symbols in $\mathcal{X} \cup \{1\}$. It should ouptut the **binary** arithmetic code for this sequence.
- You should write a function `arithmeticDecode` that takes in as arguments p and an encoded binary code and output the corresponding sequence. The decoder should terminate when it reads the symbol "1".

Remark 1. *Recall that in arithmetic coding, each sequence is represented as an interval in* $[0, 1]$. *While this works in infinite precision, you may run into numerical issues for large sequences. A finite precision encoder and decoder will appropriately rescale the intervals.*

You need to submit,

(a) The outputs of the results for the sequences given in the starter code. The code will print them out for you so you can just attach a screenshot.

(b) A printed version of your code attached to your solution.

4. **Sufficient Statistics** Let $X^n = X_1, \dots, X_n \sim P_\theta = \text{Unif}(0, \theta)$. And recall that $T(X^n) = \max_i X_i$ is a sufficient statistic.

(a) What is the distribution of $P(X|T(X^n))$?

(b) Argue that generating additional samples according to $P(X|T(X^n))$ does not help us get more information or improve our estimate of the parameter θ.