Math Foundations of ML, Fall 2018

Homework #10

Due Monday December 3, at the beginning of class

As stated in the syllabus, unauthorized use of previous semester course materials is strictly prohibited in this course.

0. Please fill out the CIOS course evaluation survey: http://b.gatech.edu/cios

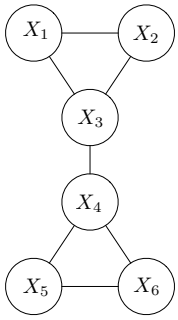
1. Using you class notes, prepare a 1-2 paragraph summary of what we talked about in class in the last week. I do not want just a bulleted list of topics, I want you to use complete sentences and establish context (Why is what we have learned relevant? How does it connect with other things you have learned here or in other classes?). The more insight you give, the better.

2. Start by reading Notes 27 on optimization using gradient descent, paying special attention to the last section "Interior point iterations".

   Suppose that we observe iid random vectors $X_1, X_2, \ldots, X_N$ in $\mathbb{R}^6$ distributed as

   $$X_n \sim \text{Normal}(\mathbf{0}, \boldsymbol{R}),$$

   where $\boldsymbol{R}$ is unknown except for the inverse covariance structure indicated by this graph:

   (a) Write down the (contrained) optimization problem that needs to be solved to get the MLE for $\boldsymbol{R}$. (There is a very similar example on page 33 of the notes.)

   (b) Write code that finds the MLE for the data vectors in the file `hw09p2data.mat`. (That file contains a 6×1000 matrix `X` whose columns are the X_n referred to above.) Compare your answer to the sample covariance (i.e. the MLE in the unconstrained case).

3. Let $X_1, X_2, \ldots$ be independent Gaussian random variables with mean 0 and variance 1. Let

   $$Z_M = \max_{1 \le m \le M} |X_m|.$$

Using Monte Carlo simulation, estimate $\mathrm{E}[Z_M]$ for $M = 1, 2, 5, 10, 20, 50, 100, \ldots, 10^5, 2 \cdot 10^5, 5 \cdot 10^5, 10^6$. Turn in a plot of $\mathrm{E}[Z_M]$ versus M on appropriately scaled (log) axes.

4. Suppose that the coupled random variables $(X, Y) \in \mathbb{R} \times \{0, 1\}$ have joint distribution specified by

$$\mathrm{P}(Y = 0) = 0.4, \quad X|Y = 0 \sim \text{Normal}(-1, 4), \quad X|Y = 1 \sim \text{Normal}(1, 4).$$

We will consider the following set of classifiers for predicting Y from an observation of X:

$$\mathcal{H} = \{h_\theta(x),\, \theta \in [-10, 10]\}, \quad \text{where} \quad h_\theta(x) = \begin{cases} 0, & x < \theta, \\ 1, & x \geq \theta. \end{cases}$$

In this case, because we have been told the distribution, we can compute the true risk for every $h_\theta \in \mathcal{H}$:

$$R(h_\theta) = \mathrm{P}(Y = 1) \int_{-\infty}^{\theta} f_X(x|Y = 1)\, \mathrm{d}x \; + \; \mathrm{P}(Y = 0) \int_{\theta}^{\infty} f_X(x|Y = 0)\, \mathrm{d}x. \qquad (1)$$

(In MATLAB/Python, you can compute the above with the help of the `normcdf`/`norm.cdf` command.)

(a) Write code that generates N (independent) realizations of (X, Y) then plots the empirical risk function $\hat{R}_N(h_\theta)$ overlaid on top of $R(h_\theta)$. Turn in plots of three realizations each for $N = 10, 100, 1000$. These plots should have a horizontal axis indexed by $\theta \in [-10, 10]$ (and this interval should be discretized to 1000 points).

(b) Using Monte Carlo simulation, estimate $\mathrm{E}[|R(h_\theta) - \hat{R}_N(h_\theta)|]$ for the particular case of $\theta = 0.8$ and $N = 10, 100, 1000$. Here, the expectation is with respect to the draw of the data. For a fixed N, a single experiment consists of drawing $x_1, \ldots, x_N$, computing $\hat{R}_N(h_{0.8})$, and then $|R(h_{0.8}) - \hat{R}_N(h_{0.8})|$ (the quantity $R(h_{0.8})$ is deterministic). Run this experiment many times and average the results to get your estimate. Then repeat for the other values of N.

(c) Using Monte Carlo simulation, estimate

$$\mathrm{E}\left[\max_{h_\theta \in \mathcal{H}} |R(h_\theta) - \hat{R}_N(h_\theta)|\right]$$

for $N = 10, 100, 1000$. As above, the expectation is with respect to the random draw of the data $x_1, \ldots, x_N$, so your simulation framework should be similar. The main difference is that every experiment produces a random *function* $\hat{R}_N(h_\theta)$ of θ that is compared against the deterministic function $R(h_\theta)$. You can compute the max by gridding the θ axis at sufficiently many points.

(d) Using Monte Carlo simulation, estimate the average performance (generalization error) $\mathrm{E}[R(\hat{h}_N)]$ of the empirical risk minimizer

$$\hat{h}_N = \arg\min_{h_\theta \in \mathcal{H}} \hat{R}_N(h_\theta),$$

for $N = 10, 100, 1000$. (You again need simulations as above to generate the $\hat{h}_N$ — given the minimizer, computing $R(\hat{h}_N)$ can be done with (1).) As before,

$\hat{h}_N$ is a random classification rule (because of the randomness of the data), and so $R(\hat{h}_N)$ is a random number, even though $R(\cdot)$ is a deterministic function. Compare your estimate of $\mathrm{E}[R(\hat{h}_N)]$ to the risk of the Bayes classifier $R(h_{\text{bayes}})$, where as usual

$$h_{\text{bayes}} = \arg\min_{h_\theta \in \mathcal{H}} R(h_\theta).$$

5. The file `hw10p5data.mat` contains an array `X` whose columns should be interpreted as data points in $\mathbb{R}^2$. Implement the EM algorithm, and use it to train a Gaussian mixture model with 5 components. Initialize the algorithm with densities of the form $\text{Normal}(\boldsymbol{m}_k, \gamma_k \mathbf{I})$ for reasonable choices of $\boldsymbol{m}_k$ and γ_k that you surmise simply by inspecting a scatter plot by eye. Turn in your code, and contour plots of each of your 5 mixture components overlayed on a scatter plot of the data.