Math Foundations of ML, Fall 2017

Homework #8

Due Monday November 13, at the beginning of class

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

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. Let

$$\boldsymbol{A} = \begin{bmatrix} 2 & 4 & -1 \\ 1 & -2 & 1 \\ 4 & 0 & 1 \\ 5 & 6 & -1 \\ 8 & -4 & 2 \end{bmatrix}.$$

Suppose that we observe

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$$

where

$$\boldsymbol{y} = \begin{bmatrix} 6.1709 \\ -1.6492 \\ 6.6345 \\ 13.8419 \\ 4.9064 \end{bmatrix},$$

and $\boldsymbol{e}$ has covariance matrix

$$\boldsymbol{R} = \begin{bmatrix} 1 & 1/3 & 1/9 & 1/27 & 1/81 \\ 1/3 & 1 & 1/3 & 1/9 & 1/27 \\ 1/9 & 1/3 & 1 & 1/3 & 1/9 \\ 1/27 & 1/9 & 1/3 & 1 & 1/3 \\ 1/81 & 1/27 & 1/9 & 1/3 & 1 \end{bmatrix}.$$

(a) Find the best linear unbiased estimate $\hat{\boldsymbol{x}}_{\text{blue}}$ of $\boldsymbol{x}$.

(b) What is the mean squared error of your estimate $\mathrm{E}[\|\boldsymbol{x} - \hat{\boldsymbol{x}}_{\text{blue}}\|_2^2]$?

3. Suppose that we want to create a realization of Gaussian noise $\boldsymbol{e} \in \mathbb{R}^5$ with covariance matrix as in the last problem. We have at our disposal a random number generator that creates independent and identically distributed Gaussian random variables with variance 1. We use this to generate $\boldsymbol{e}_{\text{ind}}$, and then pass the output through a matrix to give it the desired covariance structure. Find a matrix $\boldsymbol{Q}$ such that the covariance matrix of $\boldsymbol{Q}\boldsymbol{e}_{\text{ind}}$ is $\boldsymbol{R}$.

4. Let X be a Gaussian random vector of length D,

$$X \sim \text{Normal}(0, \boldsymbol{R}).$$

We observe the first p entries of X, while leaving the last $D-p$ entries unobserved. Call the observed variables X_o and the hidden variables X_h. Before the observation, our "best guess" for the hidden variables is simply their mean, i.e. $\mathbf{0}$; the mean-squared error (MSE) of this guess is

$$\text{E}[\|X_h - \mathbf{0}\|_2^2] = \text{E}[\|X_h\|_2^2].$$

Given an observation $X_o = \boldsymbol{x}_o$, our best guess for the hidden variable is the conditional mean:

$$\hat{\boldsymbol{\mu}}_{h|o} = \text{E}[X_h | X_o = \boldsymbol{x}_o].$$

Show that no matter what the observed values $\boldsymbol{x}_o$ are, the MSE of your best guess decreases:

$$\text{E}[\|X_h - \hat{\boldsymbol{\mu}}_{h|o}\|_2^2 | X_o = \boldsymbol{x}_0] \leq \text{E}[\|X_h\|_2^2].$$

5. Let X_1, X_2, and Z be independent Gaussian random variables (scalars) with mean 0 and variance 1. Set

$$X_3 = X_1/3 + X_2/3 + Z/3.$$

   (a) Write down the covariance matrix for the Gaussian random vector $\begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix}$. Also compute the inverse covariance $\boldsymbol{S} = \boldsymbol{R}^{-1}$.

   (b) Suppose I observe $X_3 = x_3$. Conditioned on this observation, are X_1 and X_2 still independent? How is your answer manifest in the structure of the inverse covariance matrix?

6. Let X be a Gaussian random vector,

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_D \end{bmatrix} \sim \text{Normal}(\mathbf{0}, \boldsymbol{R}).$$

We say that entries X_i, X_j are *conditionally independent* if observing all of the other entries in the vector makes X_i, X_j independent. That is, if

$$X_{\overline{(i,j)}} = \{X_k \;:\; k \neq i, k \neq j\}$$

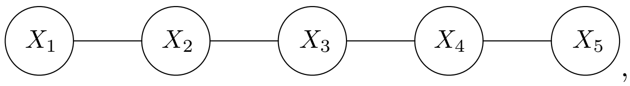
is the collection of $D-2$ entries in X that are not X_i or X_j, given any observation $X_{\overline{(i,j)}} = \boldsymbol{v}$, the random variables $X_i | X_{\overline{(i,j)}} = \boldsymbol{v}$ and $X_j | X_{\overline{(i,j)}} = \boldsymbol{v}$ are independent:

$$f_{X_i,X_j}(x_i, x_j | X_{\overline{(i,j)}} = \boldsymbol{v}) = f_{X_i}(x_i | X_{\overline{(i,j)}} = \boldsymbol{v}) \cdot f_{X_j}(x_j | X_{\overline{(i,j)}} = \boldsymbol{v}).$$

As we discussed in class, we can immediately see if X_i and X_j are conditionally independent by looking at the inverse covariance $\boldsymbol{S} = \boldsymbol{R}^{-1}$, as $S[i,j] = 0$ if and only if X_i and X_j are conditionally independent.

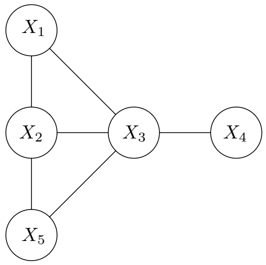
Another way to summarize the conditional independence structure is using a graph. Each of the nodes of the graph corresponds to an entry X_i, and there is an edge between node i and node j if X_i and X_j are conditionally dependent (i.e. not conditionally independent). Equivalently, if there is not an edge between node i and node j, the corresponding entry of the inverse covariance will be zero.

For example, as we saw in the example in class, the graph on the left below has the inverse covariance structure on the right:

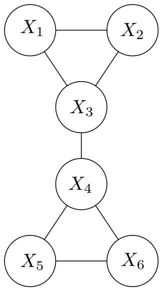


$$, \qquad \boldsymbol{S} = \begin{bmatrix} x & x & 0 & 0 & 0 \\ x & x & x & 0 & 0 \\ 0 & x & x & x & 0 \\ 0 & 0 & x & x & x \\ 0 & 0 & 0 & x & x \end{bmatrix}.$$

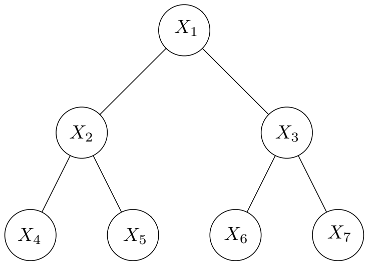
(a) For each of the graphs below, indicate the inverse covariance structure



(i)



(ii)



(iii)

(b) Suppose that removing vertex X_k separates the graph into two connected components X_{c_1} and X_{c_2} so that there is no path between any vertex in X_{c_1} and X_{c_2}. For example, if we removed X_3 in (ii) above, we could take

$$X_{c_1} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \quad \text{and} \quad X_{c_2} = \begin{bmatrix} X_4 \\ X_5 \\ X_6 \end{bmatrix}.$$

Argue that after observing $X_k = x_k$ at such a node, any $X_i \in X_{c_1}$ and any $X_j \in X_{c_2}$ will be independent; that is

$$f_{X_i,X_j}(x_i, x_j | X_k = x_k) = f_{X_i}(x_i | X_k = x_k) \cdot f_{X_j}(x_j | X_k = x_k).$$

7. *The log-barrier technique for constrained optimization.*

(a) Make an illustrative plot the function $\phi_\tau(t) = -\tau \log(t)$ on $\{t > 0\}$ for $\tau = 2, 1, 1/2, 1/5$ (on the same axes). What is happening as $\tau \to 0$?

(b) Show that $\phi_\tau(t)$ is convex on $\{t > 0\}$ for all $\tau > 0$. (The function is clearly differentiable, so it is enough to show that the second derivative is non-negative.)

(c) Let $f_1(\boldsymbol{x}), \ldots, f_M(\boldsymbol{x})$, $f_m : \mathbb{R}^D \to \mathbb{R}$, be convex functions on $\mathbb{R}^D$, and let $b_1, \ldots, b_m$ be arbitrary real numbers. Show that

$$\Phi_\tau(\boldsymbol{x}) = -\tau \sum_{m=1}^{M} \log(b_m - f_m(\boldsymbol{x}))$$

is a convex function on the region $\{\boldsymbol{x} : f_m(\boldsymbol{x}) < b_m,\ m = 1, \ldots, M\} \subset \mathbb{R}^D$. (Hint: $\log(\cdot)$ is also monotonic, and it is easy to show that the sum of convex functions is again convex.)

All of the above gives us a way to turn an optimization program with *convex constraints* into a series of unconstrained problems that can be solved in any number of ways (using gradient descent, for example). If we have the abstract program

$$\begin{aligned} \underset{\boldsymbol{x} \in \mathbb{R}^D}{\text{minimize}}\ f_0(\boldsymbol{x}) \quad \text{subject to} \quad & f_1(\boldsymbol{x}) \le b_1 \\ & f_2(\boldsymbol{x}) \le b_2 \\ & \vdots \\ & f_M(\boldsymbol{x}) \le b_M, \end{aligned} \tag{1}$$

where all of the f_m are convex, then we can approximate its solution by solving

$$\underset{\boldsymbol{x} \in \mathbb{R}^D}{\text{minimize}}\ f_0(\boldsymbol{x}) - \tau \sum_{m=1}^{M} \log(b_m - f_m(\boldsymbol{x})) \tag{2}$$

for some small τ. That is, if $\hat{\boldsymbol{x}}$ is the solution to (1) and $\hat{\boldsymbol{x}}_\tau$ is the solution to (2), then $\hat{\boldsymbol{x}}_\tau \to \hat{\boldsymbol{x}}$ as $\tau \to 0$. In optimization, this is called the *log barrier technique.*

(d) Suppose we have a set of linear constraints $\boldsymbol{C}\boldsymbol{x} \le \boldsymbol{b}$. This would be the same as taking $f_m = \boldsymbol{c}_m \mathrm{T} \boldsymbol{x}$ above, where $\boldsymbol{c}^{\mathrm{T}}$ is the mth row of $\boldsymbol{C}$ (and b_m is the mth entry in $\boldsymbol{b}$). Compute the gradient of

$$\Phi(\boldsymbol{x}) = -\tau \sum_{m=1}^{M} \log(b_m - \boldsymbol{c}_m^{\mathrm{T}} \boldsymbol{x}).$$

(e) Write a script that computes the solution to the non-negative least-squares problem

$$\underset{\boldsymbol{x} \in \mathbb{R}^D}{\text{minimize}}\ \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \quad \text{subject to} \quad \boldsymbol{x} \ge \boldsymbol{0}$$

using the log-barrier technique. Use it to find the particular solution when

$$\boldsymbol{A} = \begin{bmatrix} 1 & -1 & -1 & 0 \\ 2 & 0 & 3 & 0 \\ -2 & 0 & 1 & 1 \\ 1 & 4 & 0 & 1 \\ 0 & 3 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 2 \\ 0 \end{bmatrix}.$$

Find the solution $\hat{\boldsymbol{x}}_\tau$ for $\tau = 1, 1/2, 1/4$ (you will want to use the solution for one value of τ to initialize gradient descent for the next smaller value). Keep cutting τ in half until you have a good idea what the solution $\hat{\boldsymbol{x}}$ is.

A few notes:

- You have to be a little careful in computing the step size α_k at each iteration; this is trickier than it is for least squares. Here is what you might do: At point $\boldsymbol{x}^{(k)}$, compute the direction you want to move (the negative gradient) $\boldsymbol{d}^{(k)}$. Start with $\alpha_k = 1$, check that $\boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)} \geq 0$, and if it is not, keep cutting α_k in half until it is. Then check that $f(\boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}) \leq f(\boldsymbol{x}^{(k)})$, and if it is not, keep cutting α_k in half until it is[1].
- For each τ, you should run gradient descent until the maximum entry in gradient, $\|\nabla f(\boldsymbol{x}^{(k)}\|_\infty$ is below some small threshold (say 10^{-4}) or some large number (say $20,000$) of iterations have passed.

[1] Here $f(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \Phi_\tau(\boldsymbol{x})$.