Math Foundations of ML, Fall 2018

Homework #6

Due Friday October 19, 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. Recall the Gram-Schmidt algorithm from HW 3: if $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N\}$ is a set of linearly independent vectors in $\mathbb{R}^M$ (so clearly $N \leq M$) then we can generate a sequence of orthonomal vectors $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_N\}$ such that

$$\text{span}(\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N\}) = \text{span}(\{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N\})$$

using

$$\boldsymbol{q}_1 = \frac{\boldsymbol{a}_1}{\|\boldsymbol{a}_1\|_2}$$

then for $k = 2, 3, \ldots$,

$$\boldsymbol{w}_k = \boldsymbol{a}_k - \sum_{\ell=1}^{k-1} \langle \boldsymbol{a}_k, \boldsymbol{q}_\ell \rangle \boldsymbol{q}_\ell$$
$$\boldsymbol{q}_k = \frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|_2}.$$

   (a) As a warm-up and review, find $\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3$ when

$$\boldsymbol{a}_1 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \\ -1 \end{bmatrix}, \quad \boldsymbol{a}_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \boldsymbol{a}_3 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \end{bmatrix}.$$

   Feel free to use a computer to do the calculations; just explain what you did in your write-up.

   (b) For $\{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3\}$ and $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3\}$ as in the last part, let

$$\boldsymbol{A} = \begin{bmatrix} | & | & | \\ \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 \\ | & | & | \end{bmatrix}, \quad \boldsymbol{Q} = \begin{bmatrix} | & | & | \\ \boldsymbol{q}_1 & \boldsymbol{q}_2 & \boldsymbol{q}_3 \\ | & | & | \end{bmatrix}.$$

   Show how we can write $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{R}$, where $\boldsymbol{R}$ is upper triangular. Do this by explicitly calculating $\boldsymbol{R}$. (Hint: just keep track of your work while doing part (a)).

(c) Suppose I run the algorithm above on a general $M \times N$ matrix $\boldsymbol{A}$ with linearly independent columns (full column rank). Explain how the Gram-Schmidt algorithm can be interpreted as finding a $M \times N$ matrix $\boldsymbol{Q}$ with orthonormal columns and an upper triangular matrix $\boldsymbol{R}$ such that $\boldsymbol{A} = \boldsymbol{QR}$. Do this by explicitly writing what the entries of $\boldsymbol{R}$ are in terms of the quantities that appear in the algorithm. This is called the *QR decomposition* of $\boldsymbol{A}$.

(d) Prove or disprove: an upper triangular matrix is invertible if and only none of the elements along the diagonal are zero. Why does the linear independence of the columns of $\boldsymbol{A}$ mean that all of the entries along the diagonal of $\boldsymbol{R}$ will be nonzero?

(e) Suppose that an $M \times N$ matrix $\boldsymbol{A}$ has full column rank. Show that the solution to the least-squares problem

$$\underset{\boldsymbol{x} \in \mathbb{R}^N}{\text{minimize}} \ \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$$

is $\hat{\boldsymbol{x}} = \boldsymbol{R}^{-1}\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{y}$, where $\boldsymbol{A} = \boldsymbol{QR}$ is the QR decomposition of $\boldsymbol{A}$.

3. As we have seen (i.e. Homework 1, Problem 5), a function $f : \mathbb{R} \to \mathbb{R}$ of one variable that is twice differentiable is convex if its second derivative is positive everywhere. A function $f : \mathbb{R}^M \to \mathbb{R}$ of M variables is convex if every function of one variable formed by looking along a ray starting from an arbitrary point has a second derivative that is positive anywhere. That is, the function

$$g_{\boldsymbol{x},\boldsymbol{v}}(t) = f(\boldsymbol{x} + t\boldsymbol{v})$$

is a convex function of one variable (t) for all $\boldsymbol{x}, \boldsymbol{v}$.

Let $\boldsymbol{H}$ be a symmetric matrix. Show that

$$f(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x}$$

is convex if and only if $\boldsymbol{H}$ is symmetric positive semi-definite. That is, all the eigenvalues of $\boldsymbol{H}$ are non-negative, or equivalently $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{w} \geq 0$ for all $\boldsymbol{w} \in \mathbb{R}^M$.

4. Let

$$\boldsymbol{H} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} -1 \\ -3 \end{bmatrix},$$

and let $f : \mathbb{R}^2 \to \mathbb{R}$ be

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x} - \boldsymbol{b}^{\mathrm{T}}\boldsymbol{x}.$$

(a) What is the smallest value that f takes on $\mathbb{R}^2$? At what $\boldsymbol{x}_\star$ does it achieve this minimum value?

(b) Write $f(\boldsymbol{x})$ out as a quadratic function in x_1, x_2 where $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. In other words, fill in the blanks below

$$f(\boldsymbol{x}) = \_\_x_1^2 + \_\_x_2^2 + \_\_x_1x_2 + \_\_x_1 + \_\_x_2.$$

(c) Using MATLAB or Python, make a contour plot of $f(\boldsymbol{x})$ around its minimizer in $\mathbb{R}^2$. Compute the eigenvectors and eigenvalues of $\boldsymbol{H}$, and discuss what role they are playing in the geometry of your sketch.

(d) On top of the contour plot, trace out the first four steps of the gradient descent algorithm starting at $\boldsymbol{x}_0 = \boldsymbol{0}$.

5. A look at the plots on pages 119 and 123 as well as the plot you made in answering the previous question should give you the following intuition: if the initial point $\boldsymbol{x}_0$ is along one of the axes of the ellipses defined by the contour lines of $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{H}\boldsymbol{x} - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{b}$, then steepest descent converges in one step, i.e. $\boldsymbol{x}_1 = \boldsymbol{x}_\star$.

   We will now make this intuition formal. Suppose

$$\boldsymbol{x}_0 = \boldsymbol{x}_\star + \beta\boldsymbol{v},$$

   where $\boldsymbol{v}$ is an eigenvector of $\boldsymbol{H}$ with associated eigenvalue λ. Show that

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \alpha_0\boldsymbol{r}_0 = \boldsymbol{x}_\star,$$

   where $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{H}\boldsymbol{x}_0$ and α_0 is chosen as on page 121 of the notes.

6. Write a MATLAB function `sdsolve` that implements the steepest descent algorithm. The function should be called as

```
[x, iter] = sdsolve(H, b, tol, maxiter);
```

   where `H` is a $N \times N$ symmetric positive definite matrix, `b` is a vector of length N, and `tol` and `maxiter` specify the halting conditions. Your algorithm should iterate until $\|r_k\|_2/\|b\|_2$ is less than `tol` or the maximum number of iterations `maxiter` has been reached. For the outputs: `x` is your solution, and `iter` is the number of iterations that were performed. Run your code on the $\boldsymbol{H}$ and $\boldsymbol{b}$ in the file `hw6p6_data.mat` for a `tol` of 10^{-6}. Report the number of iterations needed for convergence, and for your solution $\hat{\boldsymbol{x}}$ verify that $\boldsymbol{H}\hat{\boldsymbol{x}}$ is within the specified tolerance of $\boldsymbol{b}$.