EE 516: Mathematical Foundations of Machine Learning **Winter 2023**

Homework 5
Due: February 17, 2023, 11:59PM PT

Student Name: *Instructor Name: John Lipor*

Problem 1 (5 pts, 3 pts)

Let $A \in \mathbb{R}^{m \times n}$ with $m \geq n$ and $\text{rank}(A) = n$, i.e., A has full column rank.

(a) Show using an SVD of A that $A^+ = (A^T A)^{-1} A^T$. When A is known to have full column rank, one could use this direct formula to compute the pseudoinverse instead of employing an SVD.

(b) Deduce that the orthogonal projection onto the space spanned by the columns of A (i.e., the *range* of A) is given by $P = A(A^T A)^{-1} A^T$ using the SVD (i.e., do not just show that $P = AA^+$).

Problem 2 (6 pts, 2 pts, 5 pts)

Consider the plane $ax + by + cz = 0$.

(a) Describe how you would use an SVD to find basis vectors for the plane.

(b) How many basis vectors are required to express a point on the plane?

(c) Find the point on the plane that is closest to (α, β, γ).

Problem 3 (3 pts)

In the usual least-squares problem, we found the minimum-norm x that minimized $\|r\|_2$, where $r = Ax - b$ is the residual. We saw that the optimal x can be expressed in terms of b and an SVD of A. Let A be an $m \times n$ matrix so that x is an $n \times 1$ vector and b is an $m \times 1$ vector. Thus r is an $m \times 1$ vector. In an upcoming application known as iteratively reweighted least-squares (IRLS), we will instead minimize $\|Wr\|_2$, where W is a diagonal *weight* matrix; this is referred to as the *weighted least-squares* problem. Determine the optimal x for the weighted least-squares problem

$$\hat{x} = \arg\min_x \|W(Ax - b)\|_2^2 .$$

Your answer should be in terms of the pseudoinverse or SVD of a new matrix that you define—the dependence on W, A, and b need not be directly apparent, and the solution should be *very* simple.

Problem 4 (5 pts each)

(a) Determine the solution to the problem below, either in terms of the pseudoinverse of an appropriately defined matrix or of a set of equations resembling the normal equations.

$$\hat{x} = \arg\min_x \|Ax - b\|_2^2 + \|Cx - d\|_2^2 + \lambda \|x\|_2^2 .$$

(b) Assume $A \in \mathbb{R}^{m \times n}$ with $m > n$ has rank n. Find the solution of

$$\hat{x} = \arg\min_x \|Ax - b\|_2^2 + \|Ax + b\|_2^2 .$$

(c) Does the answer to part (b) change if $A \in \mathbb{R}^{m \times n}$ with $m > n$ but with rank less than n? Justify your answer using the SVD/pseudoinverse viewpoint of the least-squares solution.

Problem 5 (8 pts)

Let $X = U\Sigma V^T$ be a rank-r matrix. Suppose we are given the estimates $\hat{U}$ and $\hat{V}$ (of U and V, respectively) formed from the SVD of the noisy matrix $\hat{X} = X + N$, where N is a noise matrix of the same size as X. Consider the diagonal matrix $\hat{D}$ that arises as the solution to the optimization problem

$$\hat{D} = \underset{D=\mathrm{diag}(d_1,\ldots,d_r)}{\arg\min} \left\| X - \hat{U}(:,1:r) D \hat{V}(:,1:r)^T \right\|_F ,$$

where the notation $A(:,1:r)$ denotes the first r columns of the matrix A. Express $\hat{D}$ in closed form in terms of $U, V, \Sigma, r, \hat{U}$ and $\hat{V}$. **Hint:** Follow a similar approach to that used on page 6.5 of the notes.

Problem 6 (4 pts each)

In this problem, you will explore the differences between low-rank approximation (PCA) and least-squares approximation.

(a) First consider one-dimensional data lying with the affine relationship $y = ax + b$. Using the script `prob6a`, fit the given data in `xx,yy` using a least-squares estimate for a, b and plot the resulting line. Next, fit the same data using a rank-1 approximation, as described on Pg. 6.12-6.13 of the notes.
Turn in the resulting plot.
Hint: To obtain a correct low-rank approximation, you must first *center* the data by removing its mean. This can be done efficiently using the `mean` and `repmat` functions. Don't forget to add the mean back when you go to plot.

(b) Next, we'll fit a two-dimensional plane using each method. Complete the least-squares and low-rank approximations in `prob6b`. Although it is difficult to see, the resulting planes are different. The takeaway here is that we aren't limited to fitting one-dimensional data!
Turn in the resulting plot.
Note: There is no offset for this part, so no need to center.

Problem 7 (5 pts)

In many applications, we want to fit a system of equations and know that the variables being fit are non-negative. For example, if we are trying to fit $F = ma$, we know that m and a are positive. In this case, we wish to solve the *non-negative least-squares* problem

$$\begin{aligned} x_{\text{NNLS}} = &\quad \arg\min_x \|Ax - b\|_2^2 \\ \text{subject to} &\quad x \geq 0, \end{aligned}$$

where we have explicitly imposed the constraint that the solution must be non-negative. If we solve the least-squares problem without the non-negativity constraint and the solution happens to be non-negative, then that solution is also the optimal non-negative solution. However, typically the unconstrained solution will have negative values. Setting the negative values to zero after the fact usually does not solve the stated problem.

We can solve optimization problems of the form

$$\begin{aligned} x_{\text{opt}} = &\quad \arg\min_x \|Ax - b\|_2^2 \\ \text{subject to} &\quad x \in \mathcal{C}, \end{aligned}$$

where $\mathcal{C}$ denotes a convex set, via a projected gradient method that combines the usual gradient descent update with an additional projection onto the convex set. For the non-negative least-squares problem, the update iteration is given by

$$x_{k+1} = P_{\mathcal{C}}\left(x_k - \mu A^H(Ax_k - b)\right),$$

and it will converge to the correct solution for $0 < \mu < 2/\sigma_1(A)^2$ with

$$P_{\mathcal{C}}(z) = \max(0, z).$$

Your task is to implement the non-negative least-squares gradient descent algorithm by completing the `nnls` file and testing it on the included `prob7` script. Test your algorithm by plotting $\|x_k - \hat{x}\|_2$ vs k as in the previous assignment **for both** your `lgsd` function from HW4 and your `nnls` function from this homework. Note: You can obtain $\hat{x}$ using the MATLAB command `lsqnonneg` or your own result after 500 iterations in PYTHON (I found the `scipy.optimize.nnls` did not work very well).

Turn in your `nnls` code and the resulting plot.

Problem 8 (5 pts)

Spend at most one hour reading the paper, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation" by Aharon, Elad, and Bruckstein again. Answer the below questions:

- Review the five bullets from Homework 1, Problem 10. Has your ability to understand any of these improved? If so, which items do you now understand better?
- Which topics of the course (e.g., matrix factorizations, least squares) have contributed most to your answer above?
- Which aspects of the course (lectures, exercises, homework, demos, exam) have contributed most to your answer above?

Course Project

The course project description has been listed on the course website. Please turn in the names of your group members and which topic your group has selected.