Homework 3, CS 4220, Spring 2021
Instructor: Austin R. Benson
Due Friday, March 26, 2021 at 3:44pm ET on CMS (before lecture)

Policies

Submission. Submit your write-up as a single PDF on CMS: `https://cmsx.cs.cornell.edu`.
Coding questions. You can use any programming language for the coding parts of the assignment. Include your code in your write-up.
Typesetting. Your write-up must be typeset with LaTeX.
Collaboration. Please discuss and collaborate on the homework, but you have to write your own solutions and code.
Resources and attribution. Feel free to use any resources that might be helpful, and provide attribution for any key ideas. We only ask that you work on the problems in earnest. Please do not hunt for solutions with a search engine.

Problems

1. *Cross validation for regularized least squares.*

   In the last homework, you solved l_2-regularized least squares problems of the form

$$\arg\min_x \|Ax - b\|_2^2 + \beta\|x\|_2^2. \tag{1}$$

   Different values of β would give different solutions. One way to find a good value of β is with *cross validation*. We first split the rows of A into two parts. Then given a candidate β, we fit x using one part (using regularization) and evaluate our solution quality on the other part. We can repeat this process many times and see which value of β performs well on average.

   One specific way to do this is the following. Let $A^{(-i)}$ be the matrix of A with row i deleted and $b^{(-i)}$ be the vector b with ith entry deleted. Let $\hat{x}^{(-i)}$ be the solution to

$$\arg\min_x \|A^{(-i)}x - b^{(-i)}\|_2^2 + \beta\|x\|_2^2. \tag{2}$$

   Then we seek to find a β to minimize

$$\sum_{i=1}^{m} (a_i^T \hat{x}^{(-i)} - b_i)^2, \tag{3}$$

   where a_i^T is the ith row of A. A common practice would be to choose several β ahead of time, pick the β^* that minimizes eq. (3), and finally solve eq. (1) with $\beta = \beta^*$.

   For this problem, assume that $A \in \mathbb{R}^{m\times n}$ is full rank with $m > n$.

   (a) What is the computational complexity of evaluating eq. (3) if we naively compute a QR factorization to solve eq. (2) for each of the m least squares problems?

   (b) Show that the normal equations for eq. (2) are given by

$$(A^TA + \beta I - a_i a_i^T)\hat{x}^{(-i)} = A^Tb - b_i a_i.$$

(c) Let $z_i = -a_i^T \hat{x}^{(-i)}$. Using part (b), show that

$$\begin{bmatrix} A^T A + \beta I & a_i \\ a_i^T & 1 \end{bmatrix} \begin{bmatrix} \hat{x}^{(-i)} \\ z_i \end{bmatrix} = \begin{bmatrix} A^T b - b_i a_i \\ 0 \end{bmatrix}.$$

(d) Let $\hat{x} = \arg\min_x \|Ax - b\|_2^2 + \beta\|x\|_2^2$ and $\ell_i^2 = a_i^T (A^T A + \beta I)^{-1} a_i$. Using part (c), show that

$$b_i - a_i^T \hat{x}^{(-i)} = \frac{(b - A\hat{x})_i}{1 - \ell_i^2}.$$

(e) Now if we can efficiently find the ℓ_i, then we can efficiently compute eq. (3). Let

$$\begin{bmatrix} A \\ \sqrt{\beta} I \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R$$

be a thin QR factorization. Show that $\ell_i^2 = \|q_i\|_2^2$, where q_i^T is the ith row of Q_1.

(f) Using the above tricks, how fast can you compute the sum in eq. (3)?

2. *Power method problems and solutions.*

Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix with eigenvalues satisfying $|\lambda_1| = |\lambda_2| > |\lambda_3| > \ldots > |\lambda_n|$. Suppose that we initialize the power method with a vector x_0 that is *not* orthogonal to any eigenvectors of A corresponding to λ_1 or λ_2.

(a) Explain how the power method can converge or not, depending on the relationship between λ_1 and λ_2.

(b) For the case where the power method does not converge, develop a modified version of the power method that can be used to compute λ_1 and λ_2.

(c) **Optional, ungraded question that can count towards class participation credit.** Give the rates of convergence in the above two casees.

3. *Shifted similarity.*

The QR algorithm with arbitrary shifts and initial tri-diagonalization of a symmetric matrix A is given as follows.

Algorithm 1 QR algorithm with tri-diagonalization and shifts

input: $A \in \mathbb{R}^{n \times n}$ symmetric
Compute an $n \times n$ orthogonal $Q^{(0)}$ so that $A^{(0)} = \left(Q^{(0)}\right)^T A Q^{(0)}$ is tri-diagonal.
for $k = 1, 2, 3, \ldots$ **do**
    Pick a shift $\mu^{(k)}$.
    Compute the QR factorization $A^{(k-1)} - \mu^{(k)} I = Q^{(k)} R^{(k)}$.
    $A^{(k)} = R^{(k)} Q^{(k)} + \mu^{(k)} I$.
end for

Show that $A^{(k)}$ is similar to A.

4. *Spectral bipartitioning.*

Suppose that we have an undirected graph $G = (V, E)$ with n nodes. Recall from class that we can represent G with a symmetric *adjacency matrix* $A \in \mathbb{R}^{n \times n}$, where $A_{ij} = 1$ if edge $(i, j) \in E$, and $A_{ij} = 0$ otherwise. In this problem, we will develop an algorithm for finding a balanced partition of a graph using eigenvectors of a matrix associated with A.

Suppose that we are looking for a set of nodes $S \subset V$ such that there are as few as possible edges going from S to the rest of the graph, i.e., we want to minimize

$$\text{cut}(S) \equiv |\{(i, j) \in E \mid i \in S, j \notin S\}|.$$

This is the famous min-cut problem, which can be solved in polynomial time.

We will add a constraint that S contains exactly half of the nodes, i.e., $|S| = |V|/2$ (we will assume that there is an even number of nodes). This makes the problem much more difficult to solve; in fact, it is NP-hard. Formally, the optimization problem is

$$\arg\min_{S \subset V} \ \text{cut}(S) \quad \text{subject to} \quad |S| = |V|/2. \tag{4}$$

(a) First, let's formulate the cut term as a matrix computation. Let D be a diagonal matrix, where $D_{ii} = \sum_j A_{ij}$, and let $L = D - A$ (this matrix is called the *graph Laplacian*). Consider a candidate set S and let $x_S \in \mathbb{R}^n$ be the vector whose ith entry equals 1 if $i \in S$ and equals -1 otherwise. Show that, for some constant c,

$$x_S^T L x_S = c \cdot \text{cut}(S).$$

(b) Show that L is positive semi-definite ($x^T L x \geq 0$ for any vector $x \in \mathbb{R}^n$) and that the vector e of all ones is an eigenvector of L with eigenvalue 0.

(c) Show that (i) x_S is orthogonal to e if and only if S satisfies the constraint in eq. (4) and (ii) $x_S^T x_S = n$ for any set S.

(d) Based on our results so far, we have shown that eq. (4) can be reformulated as

$$\arg\min_{S} \ x_S^T L x_S \quad \text{subject to} \quad x_S^T e = 0, x_S^T x_S = n.$$

The problem is still hard, but we can *relax* it to one where we find can find a solution efficiently. To do this, we will search over all vectors $x \in \mathbb{R}^n$, rather than vectors x_S:

$$\arg\min_{x \in \mathbb{R}^n} \ x^T L x \quad \text{subject to} \quad x^T e = 0, x^T x = n. \tag{5}$$

Show what the solution to eq. (5) is. After solving this relaxed problem, we can *round* it to form $S = \{i \mid x_i > 0\}$. The rounded solution S is an approximate solution and may not satisfy the constraints. However, the rounded solution can be a pretty good, and there is a lot of theory for these types of algorithms.

(e) Download the following data:

- `https://github.com/arbenson/cs4220_2021sp/blob/main/minnesota-graph.txt`
  This is a graph of a Minnesota road network, where nodes are intersections and edges are roads that go between two intersection. Each line in the file is one edge.
- `https://github.com/arbenson/cs4220_2021sp/blob/main/minnesota-coords.txt`
  These are geographic coordinates associated with the intersections. Each line in the file lists the node index, the x coordinate, and the y coordinate.

Find the rounded solution of the relaxed problem in eq. (5) for this graph. Visualize your result using the coordinates.