CS 501 - Homework 1

16:198:501

Complete each problem to the best of your ability. Your work must be your own. When problems require code, you must provide your code (which must be your own). Code should be well commented. You are welcome to discuss the problems on the message boards and with me, but again your work must be your own and demonstrate your own understanding of the problem.

1) Let $\underline{w}$ be a column vector of unknowns, $\underline{w} = (w_1, w_2, \ldots, w_d)$. **For a given vector $\underline{c}$, show that the gradient of $\underline{w}^{\mathbf{T}}\underline{c}$ is given by $\underline{c}$. For a given symmetric matrix A, show that the gradient of $\underline{w}^{\mathbf{T}}A\underline{w}$ is given by $2A\underline{w}$.**

2) Given a $D \times D$ square matrix B, we frequently want to factor or express B in terms of other matrices, for instance maybe taking the square root of B. In this problem, for a given matrix B, we want to find a (potentially non-square, non-symmetric) matrix A such that $B = A^T A$.

   - We can generate such a B by randomly generating a $k \times D$ matrix $\tilde{A}$, with each element selected as a standard normal random variable (mean 0, variance 1), and computing $B = \tilde{A}^T\tilde{A}$. From B, we would like to (potentially) recover $\tilde{A}$.
   - Consider the following approach: for any matrix A, define L as

$$L(A) = \sum_{i=1}^{D}\sum_{j=1}^{D}\left[B - A^T A\right]_{i,j}^2, \tag{1}$$

     the sum of the squares of all the differences between the elements of B and $A^T A$. We want to find A to make L as small as possible.
   - For a given matrix A, let $G(A)$ be the matrix of derivatives so that $G_{i,j} = \partial/\partial A_{i,j}\,[L]$. **Show that $G = 4A\left(A^T A - B\right)$.**
   - We can try to find A to minimize L by implementing **gradient descent**: for any given A, we can improve it by taking

$$A_{\text{new}} = A - \alpha G(A), \tag{2}$$

     where α is a small positive constant. For sufficiently small α, you'll get that $L(A_{\text{new}}) < L(A)$, and this process can be iterated to minimize L. If we can get $L(A)$ to converge to 0, then we have discovered a solution to $B = A^T A$.
   - **Generate a random B as above, taking $D = 10, k = 10$, and taking A as a random initial matrix, implement this process to show that L decreases over time to 0 - graph your results.**
   - **For a given B ($D = 10, k = 10$), do you recover the same A every time, for different initial starting points? Why or why not.**
   - Generating B randomly as above, with $D = 5, k = 10$, suppose that the 'true' value of k is forgotten. **Try to find A for different values of k. What do you notice about the loss for different k? Can you recover the 'true' dimension?**
   - Think about the relationship between the columns of A and the matrix $A^T A$, and **use this to explain the results of the previous two bullet points.**
   - **What happens if you try to take B as the identity matrix? What does the solution A represent (for any k that works)?**

- **What happens if you try to take B as the diagonal matrix of all 1s, except for the bottom right corner which is -1? What does the solution A represent (for any k that works)?**

3) Show that if M preserves norms under multiplication ($||M\underline{v}||_2 = ||\underline{v}||_2$ for all $\underline{v}$), then the columns of M must be orthonormal with respect to each other.

> Warning: Many of you will be tempted to start this problem by saying, I'll just take the columns to be orthonormal, and then show that the norm is preserved. *This is incorrect.* For this problem, I want you to start with the assumption that norms are preserved, and then show that the columns must be orthonormal. As a hint: if the norms are preserved for all $\underline{v}$, are there any $\underline{v}$ that are particularly useful to consider?

4) Consider the problem of dynamically fitting a model to data. Suppose that you are collecting data points over time, at time t receiving feature vector $\underline{x}_t$ and corresponding output value y_t. We want to model this linearly, i.e., try to fit a model of the form $y_t = \underline{x}_t^{\mathrm{T}}\underline{w}$ (here we are assuming that the intercept/constant term is subsumed into the weights and first component of the data vectors). Obviously we are unlikely to have an exact fit given enough data points, so at time T, we would like to find the vector $\underline{w}_T$ that satisfies

$$\min_{\underline{w}} \sum_{t=1}^{T} \left(\underline{x}^{\mathrm{T}}\underline{w} - y_t\right)^2. \tag{3}$$

- **Show that if $\underline{w}_T$ denotes the solution at time T of this problem, then**

$$\underline{w}_T = R_T^{-1}U_T, \text{ where } R_T = \sum_{t=1}^{T} \underline{x}_t\underline{x}_t^{\mathrm{T}}, U_t = \sum_{t=1}^{T} y_t\underline{x}_t. \tag{4}$$

- **Assume that R_T^{-1} has been computed. Express R_{T+1}^{-1} in terms of R_T^{-1} and $\underline{x}_{T+1}$.**
- Rather than re-computing a new model $\underline{w}_t$ at each timestep t, it would be better if we could *update* our previous model to reflect the new data. **Show that**

$$\underline{w}_{T+1} = \underline{w}_T + (y_{T+1} - \underline{x}_{T+1}^{\mathrm{T}}\underline{w}_T)K_{T+1}, \tag{5}$$

  for some matrix K_{T+1}, and find a nice expression for K_{T+1}.
- *This algorithm is known as Recursive Least Squares (RLS).*
- *It may be useful to familiarize yourself with the matrix inversion lemma as a convenient means of inverting slightly modified matrices.*

5) Consider the matrix

$$A = \begin{pmatrix} 1 & 0.5 \\ 0 & 1+\epsilon \end{pmatrix} \tag{6}$$

- **Find the eigenvalues and eigenvectors of A, for $\epsilon \neq 0$, taking the eigenvectors to be of unit norm.**
- **Diagonalize A in terms of these eigenvalues and eigenvectors.**
- **Taking the limit as ϵ decreases from above to 0, what happens to the diagonalization matrices? What can you conclude when $\epsilon = 0$?**