Posting Date: Saturday Sept. 7th.

You need not hand in anything. Instead, be prepared to answer any of these problems on an upcoming take-home prelim exam. You may discuss your solutions with classmates up until the time that the prelim becomes available.

1. Suppose that you are given the scalar function of a vector, $J(\underline{x}) = 0.5\ \underline{x}^{\mathrm{T}} P \underline{x} + \underline{g}^{\mathrm{T}} \underline{x}$, where $\underline{x}$ is the $n{\times}1$ vector argument of the function, P is a given $n{\times}n$ matrix, and $\underline{g}$ is a given $n{\times}1$ vector. If $\partial J/\partial \underline{x}$ is defined to be the $1{\times}n$ row vector $[\partial J/\partial x_1, \partial J/\partial x_2, \partial J/\partial x_2, \ldots, \partial J/\partial x_n]$, then prove that $\partial J/\partial \underline{x} = \underline{x}^{\mathrm{T}} P + \underline{g}^{\mathrm{T}}$. Also, if $\partial^2 J/\partial \underline{x}^2$ is defined to be the symmetric $n{\times}n$ matrix whose ijth element is $\partial^2 J/\partial x_i \partial x_j$, then prove that $\partial^2 J/\partial \underline{x}^2 = P$. Assume that P is a symmetric matrix.

   Hints: Use summation notation to represent the value of $J(\underline{x})$ in terms of the elements of $\underline{x}$, $\underline{g}$, and P. Differentiate this expression with respect to the elements of $\underline{x}$ in order to get summation formulas for the required derivatives. Show that these summation formulas are equivalent to the matrix-vector formulas that are given above for the 2 required derivative expressions.

   Note that $\partial^2 J/\partial \underline{x}^2 = \partial[(\partial J/\partial \underline{x})^{\mathrm{T}}]/\partial \underline{x}$, i.e., that the $n{\times}n$ matrix $\partial^2 J/\partial \underline{x}^2$ is the derivative of the $n{\times}1$ column vector $(\partial J/\partial \underline{x})^{\mathrm{T}}$ with respect to the $n{\times}1$ column vector $\underline{x}$. In general, given an $m{\times}1$ column vector function $\underline{v}(\underline{x})$ that depends on the $n{\times}1$ column vector $\underline{x}$, the derivative $\partial \underline{v}/\partial \underline{x}$ is defined to be the $m{\times}n$ Jacobian matrix whose ijth element is $\partial v_i/\partial x_j$.

2. Suppose that you are given the scalar quadratic form;

$$C(\underline{x}|\underline{z}) = \frac{1}{2}\begin{bmatrix}(\underline{x}-\overline{\underline{x}})^{\mathrm{T}} & (\underline{z}-\overline{\underline{z}})^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}V_{xx} & V_{xz} \\ V_{xz}^{\mathrm{T}} & V_{zz}\end{bmatrix}\begin{bmatrix}(\underline{x}-\overline{\underline{x}}) \\ (\underline{z}-\overline{\underline{z}})\end{bmatrix}$$

   where vectors $\overline{\underline{x}}$, $\underline{z}$, and $\overline{\underline{z}}$ and the matrices V_{xx}, V_{xz}, and V_{zz} are known quantities. Prove that $(\partial C/\partial \underline{x})^{\mathrm{T}} = V_{xx}(\underline{x}-\overline{\underline{x}}) + V_{xz}(\underline{z}-\overline{\underline{z}})$ and that $\partial^2 C/\partial \underline{x}^2 = V_{xx}$.

   Hints: You should be able to use your results from Problem 1. First you might want to re-express $C(\underline{x}|\underline{z})$ as a sum of scalar terms by carrying out the matrix-vector and matrix-matrix multiplications that correspond to the above block quadratic form. You should not have to resort to writing $C(\underline{x}|\underline{z})$ in terms of elemental summation formulas.

3. Given the matrices: $A = \begin{bmatrix}1 & 1 \\ 1 & (1+10^{-8})\end{bmatrix}$ and $B = \begin{bmatrix}1 & 1 \\ 1 & (1+10^{-3})\end{bmatrix}$ and the vectors:

$$\underline{y}_A = \begin{bmatrix}1 \\ (1-3\times10^{-8})\end{bmatrix} \text{ and } \underline{y}_B = \begin{bmatrix}1 \\ (1-3\times10^{-3})\end{bmatrix},$$

   solve the following equations

$$A\,\underline{x}_A = \underline{y}_A$$

$$B\,\underline{x}_B = \underline{y}_B$$

for the 2×1 vectors $\underline{x}_A$ and $\underline{x}_B$ in the following 4 ways:

i) Analytically.

ii) By using the MATLAB left division operator, $\underline{x}_A = A\backslash\underline{y}_A$, and $\underline{x}_B = B\backslash\underline{y}_B$.

Note that division by a matrix in MATLAB is accomplished, effectively, by computing the matrix's inverse and then multiplying by that inverse. Therefore, the above solution procedures are equivalent to $\underline{x}_A = A^{-1}\,\underline{y}_A$, and $\underline{x}_B = B^{-1}\,\underline{y}_B$ if A and B are square nonsingular matrices, which they are. Note that MATLAB uses QR factorization to compute the effective inverse of the matrix when an expression involving matrix division is used.

iii) By using the MATLAB left division operator as part of a least-squares solution procedure, $\underline{x}_A = (A^TA)\backslash(A^T\underline{y}_A)$, and $\underline{x}_B = (B^TB)\backslash(B^T\underline{y}_B)$.

Least-squares techniques are not needed for these problems because they are not over-determined, but least-squares still should give the correct answers, at least in theory. Differences may arise, however, due to the build-up of numerical round-off errors in the computer.

iv) By using the MATLAB matrix inversion function, inv(), as part of a least-squares solution procedure, $\underline{x}_A = \text{inv}(A^TA)*(A^T\underline{y}_A)$, and $\underline{x}_B = \text{inv}(B^TB)*(B^T\underline{y}_B)$.

Compare the four different values of $\underline{x}_A$ and $\underline{x}_B$ that you get from these four different techniques. Be sure to use the "format long" command in MATLAB so that you can see all of the digits of your numerical answers. What can you say about the various different procedures for solving the problem? What is the effect of the squaring of the matrix that occurs in the ordinary least squares solution procedure? Before you answer these questions, compute the condition numbers of A and B by using the MATLAB function cond(). Also, compute the condition numbers of A^TA and B^TB.

In discussing your answers, keep in mind that the condition number of a matrix gives an idea of whether one can invert the matrix on a finite-precision computer and get sensible results when using the computed inverse to solve a system of linear equations. If a matrix has a high condition number, then numerical round-off errors will build up a lot during the inversion process (even if the inversion algorithm is very stable numerically), and the solution of the associated system of linear equations will not be very accurate. If one is working with a computer that is executing double-precision arithmetic, i.e., that retains 16 significant digits in its real numbers, then the number of valid significant digits in the solution to a system of linear equations of the form $A\underline{x} = \underline{y}$ should be approximately $16 - \log_{10}(\text{cond}(A))$. Thus, if $\text{cond(A)} = 10^{16}$, then the computed answer $\underline{x}$ will be garbage, but if $\text{cond(A)} = 10^{10}$, then one would expect the first 6 significant digits of the computed $\underline{x}$ to be valid. Note that a condition number of infinity corresponds to a singular matrix, and a large condition number means that the matrix is close to being singular.

4. Write a Matlab function to solve the weighted least-squares problem:

   Minimize $J(\underline{x}) = 0.5*(\underline{z} - H\underline{x})^T R^{-1}(\underline{z} - H\underline{x})$

   where R^{-1} is an $m\times m$ positive definite symmetric matrix. Use the Cholesky factorization and the QR factorization, as described in class. The function's inputs must be $\underline{z}$, H, and R. Its output must be $\hat{\underline{x}}$, the weighted least-squares estimate of x.

   Solve the problem whose data is:

```
z = [  -45.1800;...
         1.7900;...
       -31.3800;...
        26.7700;...
        27.6400]

H = [   -4.9300,   -1.3100,   -1.5900;...
        13.2600,    9.7100,   30.7000;...
       -17.0800,  -11.9100,  -12.1300;...
       -24.0300,   -2.9900,  -26.9500;...
        -2.4000,   -8.7000,    9.3900]

R = [    5.9700,   -0.9200,   -1.1800,   -7.0600,   -1.7900;...
        -0.9200,    3.4500,    1.7100,   -0.6000,   -4.0500;...
        -1.1800,    1.7100,    1.1900,    0.5600,   -1.6700;...
        -7.0600,   -0.6000,    0.5600,    9.9200,    4.8500;...
        -1.7900,   -4.0500,   -1.6700,    4.8500,    6.8700]
```

   and hand in your $\hat{\underline{x}}$ vector accurate to 4 decimal places.

   Hint: you can check your solution by testing whether the norm of the violation of the first-order necessary conditions is small in some relative sense. In other words, the quantity:

   $||-H^T R^{-1}(\underline{z} - H\hat{\underline{x}})||/||-H^T R^{-1}\underline{z}||$

   should be on the order of 10^{-16}. This particular test compares the norm of the violation of the first-order necessary condition at the solution to the norm of the violation at the arbitrary non-optimal point $\underline{x} = 0$. Of course, if $\hat{\underline{x}} = 0$, then this will be a poor test.

5. Reconsider the hypothesis testing example problem presented in lecture and in the handout notes. Suppose that the two probability distributions for the two hypotheses take the following two-sided exponential forms:

$$p(x|H_0) = \frac{1}{2\sigma_A} exp(-\frac{|x|}{\sigma_A})$$

$$p(x|H_1) = \frac{1}{2\sigma_B} exp(-\frac{|x|}{\sigma_B})$$

   with $\sigma_B > \sigma_A$. Derive a new optimal Neyman-Pearson hypothesis test that uses n independent measurements of x. Given that you want to achieve a false-alarm probability of $\alpha = 0.001$ for a test that uses n = 10 samples, determine the hypothesis test statistic, its threshold value as a multiple of σ_A, and the probability of a missed detection under the assumption that $\sigma_B = 4\sigma_A$.

Hints:

If $q = \sum_{i=1}^{n} |x_i|$, then $2q/\sigma_A$ is distributed as a chi-squared distribution of degree $2n$ under hypothesis H_0, and $2q/\sigma_B$ is distributed as a chi-squared distribution of degree $2n$ under hypothesis H_1.

6. Suppose that $q = \sum_{i=1}^{n} z_i$ with the random variables z_i for i = 1,..., n independent and identically distributed with one-sided exponential probability density functions

$$p(z_i) = \begin{cases} 0 & z_i < 0 \\ \dfrac{1}{\sigma} exp(-\dfrac{z_i}{\sigma}) & 0 \le z_i \end{cases}$$

Prove that $2q/\sigma$ is distributed as a chi-squared distribution of degree $2n$.

Hints:

First use a change of variables to show that

$$p(q, z_2, z_3, ..., z_n) = \begin{cases} \dfrac{1}{\sigma^n} exp(-\dfrac{q}{\sigma}) & \text{if } q \ge \sum_{i=2}^{n} z_i,\ z_2 \ge 0,\ z_3 \ge 0, ..., \text{and } z_n \ge 0 \\ 0 & \text{otherwise} \end{cases}$$

Next, determine

$$p(q) = \int_{-\infty}^{\infty} dz_2 \int_{-\infty}^{\infty} dz_3 ... \int_{-\infty}^{\infty} dz_n p(q, z_2, z_3, ..., z_n)$$

You can use the form of $p(q, z_2, z_3, ..., z_n)$ to modify the limits of integration in this function. Afterwards you should be able to perform the required integrals one at a time.

Finally, show that the resulting expression has the desired relationship to the probability density function for a chi-squared distribution of degree $2n$.

This proof has been performed in support of the analysis that led to one of the hints for Problem 5.

Also do the following problems from Bar Shalom:

2-1, 2-3 (assume that A is a symmetric matrix), 2-5, 2-7, 2-8 (should be easy given chapter 3 and the equivalence between MAP and MMSE estimation for linear/Gaussian problems), 2-9, 2-11, 2-14 (assume the original measurements are distributed as a Gaussian)