ECE/MATH 520, Spring 2008
Exam 2: Due Session 26

Name:________________________ 75 mins.; Total 50 pts.

1. (16 pts.) The purpose of this question is to derive a recursive least-squares algorithm where we *remove* (instead of add) a data point. To formulate the algorithm, suppose we are given matrices $\boldsymbol{A}_0$ and $\boldsymbol{A}_1$ such that

$$\boldsymbol{A}_0 = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{a}_1^T \end{bmatrix},$$

where $\boldsymbol{a}_1 \in \mathbb{R}^n$. Similarly, suppose vectors $\boldsymbol{b}^{(0)}$ and $\boldsymbol{b}^{(1)}$ satisfy

$$\boldsymbol{b}^{(0)} = \begin{bmatrix} \boldsymbol{b}^{(1)} \\ b_1 \end{bmatrix},$$

where $b_1 \in \mathbb{R}$. Let $\boldsymbol{x}^{(0)}$ be the least-squares solution associated with $(\boldsymbol{A}_0, \boldsymbol{b}^{(0)})$, and $\boldsymbol{x}^{(1)}$ the least-squares solution associated with $(\boldsymbol{A}_1, \boldsymbol{b}^{(1)})$. Our goal is to write $\boldsymbol{x}^{(1)}$ in terms of $\boldsymbol{x}^{(0)}$ and the "removed" data point $(\boldsymbol{a}_1, b_1)$. As usual, let $\boldsymbol{G}_0$ and $\boldsymbol{G}_1$ be the Grammians associated with $\boldsymbol{x}^{(0)}$ and $\boldsymbol{x}^{(1)}$, respectively.

a. Write down expressions for the least-squares solutions $\boldsymbol{x}^{(0)}$ and $\boldsymbol{x}^{(1)}$ in terms of $\boldsymbol{A}_0$, $\boldsymbol{b}^{(0)}$, $\boldsymbol{A}_1$, and $\boldsymbol{b}^{(1)}$.

b. Derive a formula for $\boldsymbol{G}_1$ in terms of $\boldsymbol{G}_0$ and $\boldsymbol{a}_1$.

c. Let $\boldsymbol{P}_0 = \boldsymbol{G}_0^{-1}$ and $\boldsymbol{P}_1 = \boldsymbol{G}_1^{-1}$. Derive a formula for $\boldsymbol{P}_1$ in terms of $\boldsymbol{P}_0$ and $\boldsymbol{a}_1$. (The formula must not contain any matrix inversions.)

d. Derive a formula for $\boldsymbol{A}_0^T \boldsymbol{b}^{(0)}$ in terms of $\boldsymbol{G}_1$, $\boldsymbol{x}^{(0)}$, and $\boldsymbol{a}_1$.

e. Finally, derive a formula for $\boldsymbol{x}^{(1)}$ in terms of $\boldsymbol{x}^{(0)}$, $\boldsymbol{P}_1$, $\boldsymbol{a}_1$, and b_1. Use this and part c to write a recursive algorithm associated with successive removals of rows from $(\boldsymbol{A}_k, \boldsymbol{b}^{(k)})$.

2. (10 pts.) Use the penalty method to solve the following problem analytically:

$$\begin{array}{ll} \text{minimize} & x_1^2 + 2x_2^2 \\ \text{subject to} & x_1 + x_2 = 3. \end{array}$$

Hint: Use the penalty function $P(x) = (x_1 + x_2 - 3)^2$. The solution you find must be exact, not approximate.

3. (12 pts.) Consider a standard form LP problem. Suppose we start with an initial basic feasible solution $\boldsymbol{x}^{(0)}$ and we apply one iteration of the simplex algorithm to obtain $\boldsymbol{x}^{(1)}$.

As pointed out in class, it turns out that we can express $\boldsymbol{x}^{(1)}$ in terms of $\boldsymbol{x}^{(0)}$ as

$$\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)} + \alpha_0 \boldsymbol{d}^{(0)},$$

where α_0 minimizes $\phi(\alpha) = f(\boldsymbol{x}^{(0)} + \alpha \boldsymbol{d}^{(0)})$ over all $\alpha > 0$ such that $\boldsymbol{x}^{(0)} + \alpha \boldsymbol{d}^{(0)}$ is feasible.

a. Show that $\boldsymbol{d}^{(0)} \in \mathcal{N}(\boldsymbol{A})$.

b. As usual, assume that the initial basis is the first m columns of $\boldsymbol{A}$, and the first iteration involves inserting $\boldsymbol{a}_q$ into the basis, where $q > m$. Let the qth column of the canonical augmented matrix be $\boldsymbol{y}_q = [y_{1q}, \ldots, y_{mq}]^T$.

   Express $\boldsymbol{d}^{(0)}$ in terms of $\boldsymbol{y}_q$.

c. Show that $\boldsymbol{d}^{(0)}$ is a descent direction if and only if $r_q < 0$.

4. (12 pts.) Suppose we are given a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ and a vector $\boldsymbol{b} \in \mathbb{R}^m$ such that $\boldsymbol{b} \geq \boldsymbol{0}$. We are interested in an algorithm that, given this $\boldsymbol{A}$ and $\boldsymbol{b}$, is guaranteed to produce one of following two outputs: (1) If there exists $\boldsymbol{x}$ such that $\boldsymbol{A}\boldsymbol{x} \geq \boldsymbol{b}$, then the algorithm produces one such $\boldsymbol{x}$. (2) If no such $\boldsymbol{x}$ exists, then the algorithm produces an output to declare so.

Describe in detail how to design this algorithm based on the simplex method.