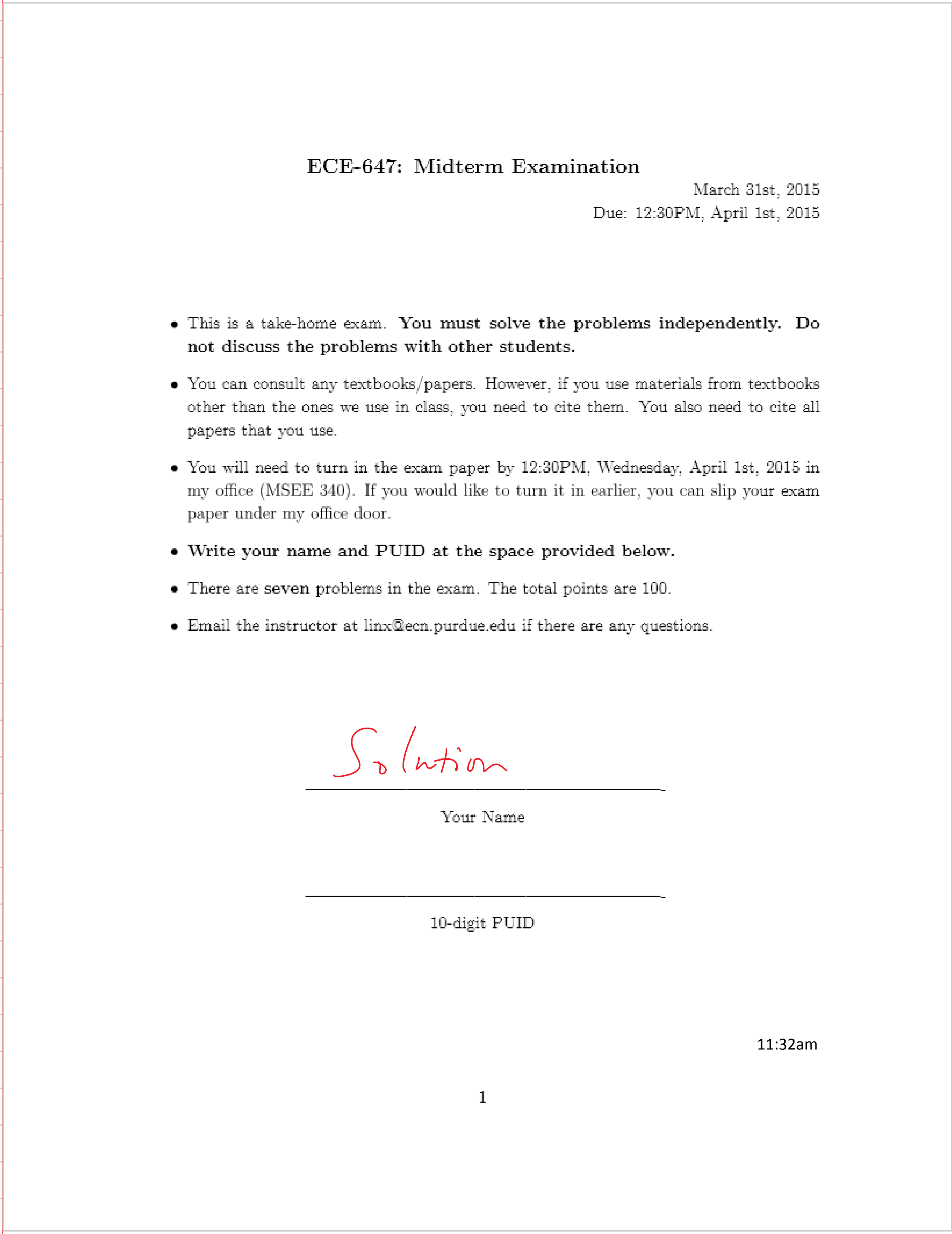
Solution 2015

Thursday, March 26, 2015
11:32 AM



ECE-647: Midterm Examination

March 31st, 2015
Due: 12:30PM, April 1st, 2015

- This is a take-home exam. **You must solve the problems independently. Do not discuss the problems with other students.**
- You can consult any textbooks/papers. However, if you use materials from textbooks other than the ones we use in class, you need to cite them. You also need to cite all papers that you use.
- You will need to turn in the exam paper by 12:30PM, Wednesday, April 1st, 2015 in my office (MSEE 340). If you would like to turn it in earlier, you can slip your exam paper under my office door.
- **Write your name and PUID at the space provided below.**
- There are **seven** problems in the exam. The total points are 100.
- Email the instructor at linx@ecn.purdue.edu if there are any questions.

Solution
____

Your Name

____

10-digit PUID

11:32am

(1) **(15 points)** (Yes or No) Is each of the following sets a convex set? **No justification is necessary.**

No (a) (3 points) The set

$$\{(x, y) \in \mathbf{R}^2 \mid x^2 + 4xy - y^2 \geq 5\},$$

Yes (b) (3 points) Suppose that X is random variable in $\mathbf{R}^n$. The set

$$\{(s, u) \mid s \in \mathbf{R}^n, u \in \mathbf{R}, \log \mathbf{E}[e^{s^T X}] \leq u\},$$

where s^T denotes the transpose of s.

Yes (c) (3 points) Let X be a real-valued random variable with $\mathbf{P}\{X = a_i\} = p_i, i = 1, ..., n$, where $a_1 < a_2 < ... < a_n$ are given real numbers. Of course, $\vec{p} = [p_1, ..., p_n] \in \mathbf{R}^n$ lies in the standard probability simplex $\{\vec{p} | \sum_{i=1}^n p_i = 1, p_i \geq 0$ for all $i\}$. The set of the probability distribution $\vec{p}$ such that $\mathbf{E}|X^3| \leq 2\mathbf{E}|X|$.

Yes (d) (3 points) Suppose that S_1 and S_2 are convex sets in $\mathbf{R}^{m \times n}$. The partial difference S defined as

$$S = \{(x, y_1 - y_2) \mid x \in \mathbf{R}^m, y_1, y_2 \in \mathbf{R}^n, (x, y_1) \in S_1, \text{ and } (x, y_2) \in S_2\}.$$

No (e) (3 points) The set

$$\{(x_1, x_2, x_3) \geq 0 | x_1 x_2 x_3 < 1\}.$$

(2) **(10 points)** In a cellular network, a mobile user may receive signals from multiple base-stations. Suppose that there are K base-stations, and base-station k is at location $x_k \in \mathbf{R}^2$. Further, suppose that all base-stations transmit signals at the common power-level P_0. If the mobile is at location y, then the signal strength received from base-station k is

$$cP_0||y - x_k||_2^{-n},$$

where $||y - x_k||_2$ denote the Euclidean distance, $c > 0$ is a constant, and n is the path-loss exponent (a constant) that is typically between 2 to 4.

Suppose that the mobile wishes to communicate with the base-station with the strongest signal. Let V denote the set of locations y such that the mobile receives a stronger signal from base-station 1 than from all other base-stations. Show that V is a convex set in $\mathbf{R}^2$. Show all intermediate steps to get full credits.

The set V can be written as

$$V = \{ y \mid cP_0 ||y - x_1||_2^{-n} \geq cP_0 ||y - X_k||_2^{-n} \text{ for all } k = 2, \cdots K \}$$

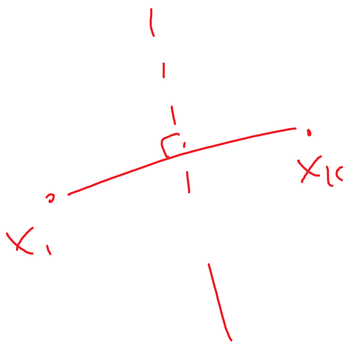
Note that for each k,

the set $\{ y \mid cP_0 ||y - X_1||_2^{-n} \geq cP_0 ||y - X_k||_2^{-n} \}$

$$= \{ y \mid ||y - X_1||_2 \leq ||y - X_k|| \}$$

which is a half space.

Therefore, V is an intersection of half-spaces, and is thus a convex set.



(3) **(15 points)** (Yes or No) Is each of the following functions a convex function? **No justification is necessary.**

Yes (a) (3 points) $f(x, y, t) = -\sqrt{xy - t^2}$, where $\mathbf{dom} f = \{x, y, t \in \mathbf{R} | xy \geq t^2\}$.

No (b) (3 points) $f(x, y) = x^2 + 4xy - 4y^2$, where $x, y \in \mathbf{R}$.

Yes (c) (3 points) $f(x, y) = x^2/(x + y)$, where $x, y \in \mathbf{R}$ and $x + y > 0$.

No (d) (3 points) Suppose that C is an arbitrary set in $\mathbf{R}^n$. The function

$$g(y) = \inf\{y^T x \mid x \in C\}$$

where $y \in \mathbf{R}^n$.

No (e) (3 points) Let X be a real-valued random variable with $\mathbf{P}\{X = a_i) = p_i, i = 1, ..., n$, where $a_1 < a_2 < ... < a_n$ are given real numbers. Of course, $\vec{p} = [p_1, ..., p_n] \in \mathbf{R}^n$ lies in the standard probability simplex $\{\vec{p} | \sum_{i=1}^n p_i = 1, p_i \geq 0$ for all $i\}$. The variance $\mathbf{Var}(X)$ as a function of the probability distribution $\vec{p}$.

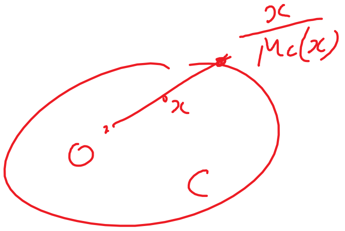
(a) $-\sqrt{xz}$ is convex, and is decreasing in each variable

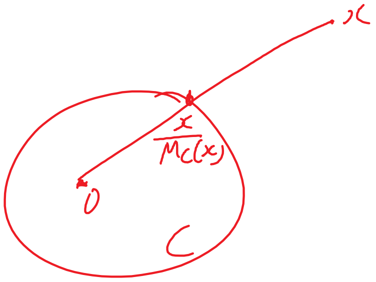
$z = y - \frac{t^2}{x}$ is concave

(4) **(15 points)** Suppose that a non-empty convex set C in $\mathbf{R}^n$ contains (as subsets) both the origin and a ball centered at the origin with some positive radius. For any $x \in \mathbf{R}^n$, define

$$M_C(x) = \inf\{t \geq 0 \mid t^{-1}x \in C\}.$$

Show that $M_c(x)$ is a convex function in x. Show all intermediate steps to get full credits.





For any $x_1, x_2 \in R^n$, let

$$z = \theta x_1 + (1-\theta) x_2$$

for some $\theta \in (0,1)$. We wish to show that

$$Mc(z) \leq \theta Mc(x_1) + (1-\theta) Mc(x_2).$$

Let us first consider $Mc(x_1)$. Note that if $\frac{x}{t_0} \in C$, then $\frac{x}{t} \in C$ for all $t \geq t_0$ because C is convex & C contains the origin. Thus, by the definition of $Mc(x_1)$, there must exist t_1 such that

$$Mc(x_1) \leq t_1 \leq Mc(x_1) + \varepsilon$$

and

$$\frac{x_1}{t_1} \in C$$

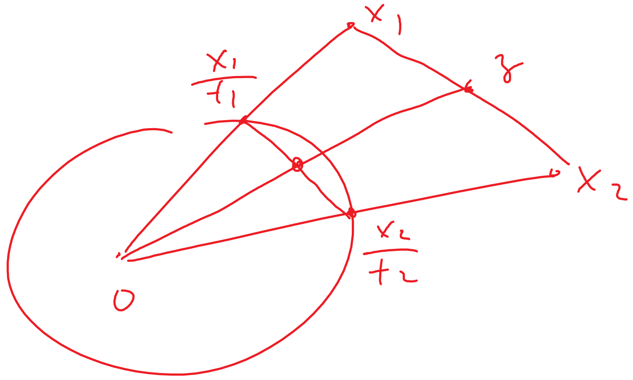
Similarly, there must exist t_2 such that

$$M_C(x_2) \le t_2 \le M_C(X_2) + \varepsilon$$

and $$\frac{X_2}{t_2} \in C.$$

Now, consider

$$\gamma = \theta x_1 + (1-\theta) X_2$$



We have

$$\gamma = \theta t_1 \frac{X_1}{t_1} + (1-\theta) t_2 \frac{X_2}{t_2}$$

$$= \left(\theta t_1 + (1-\theta) t_2\right) \left\{ \frac{\theta t_1}{\theta t_1 + (1-\theta) t_2} \frac{x_1}{t_{12}} + \frac{(1-\theta) t_2}{\theta t_1 + (1-\theta) t_2} \frac{X_2}{t_2} \right\}$$

Note that the point in the { } must also be in C since C is convex. Thus

$$M_C(\gamma) \le \theta t_1 + (1-\theta) t_2 \le \theta \cdot M_C(x_1) + (1-\theta) M_C(x_2) + \varepsilon$$

Since ε is arbitrary, we thus have

$$M_C(\gamma) \le \theta M_C(x_1) + (1-\theta) M_C(x_2)$$

(5) **(10 points)** Derive the dual problem of the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} w_i p_i \log p_i \\ \text{subject to} \quad & \sum_{i=1}^{n} p_i \leq 1, \end{aligned}$$

where $w_i, i = 1, ..., n$ are positive constants. You can assume that "log" represents the natural logarithm. Thus, due to the definition of the logarithmic function, the domain of the problem is $p_i > 0$ for all i. Show all intermediate steps to get full credits.

Associate a Lagrange multiplier to the constraint $\Sigma p_i \leq 1$. The Lagrangian is

$$L(\vec{p}, \lambda) = \sum_{i=1}^{n} w_i p_i \log p_i + \lambda\left(\sum_{i=1}^{n} p_i - 1\right)$$

$$= \sum_{i=1}^{n} \left[w_i p_i \log p_i + \lambda p_i \right] - \lambda$$

minimize each $w_i p_i \log p_i + \lambda p_i$

$$\Rightarrow \quad w_i [1 + \log p_i] + \lambda = 0$$

$$1 + \log p_i = -\frac{\lambda}{w_i}$$

$$p_i = e^{-\frac{\lambda}{w_i} - 1}$$

Thus, the dual objective value is

$$D(\lambda) = L(\vec{p}^*, \lambda)$$

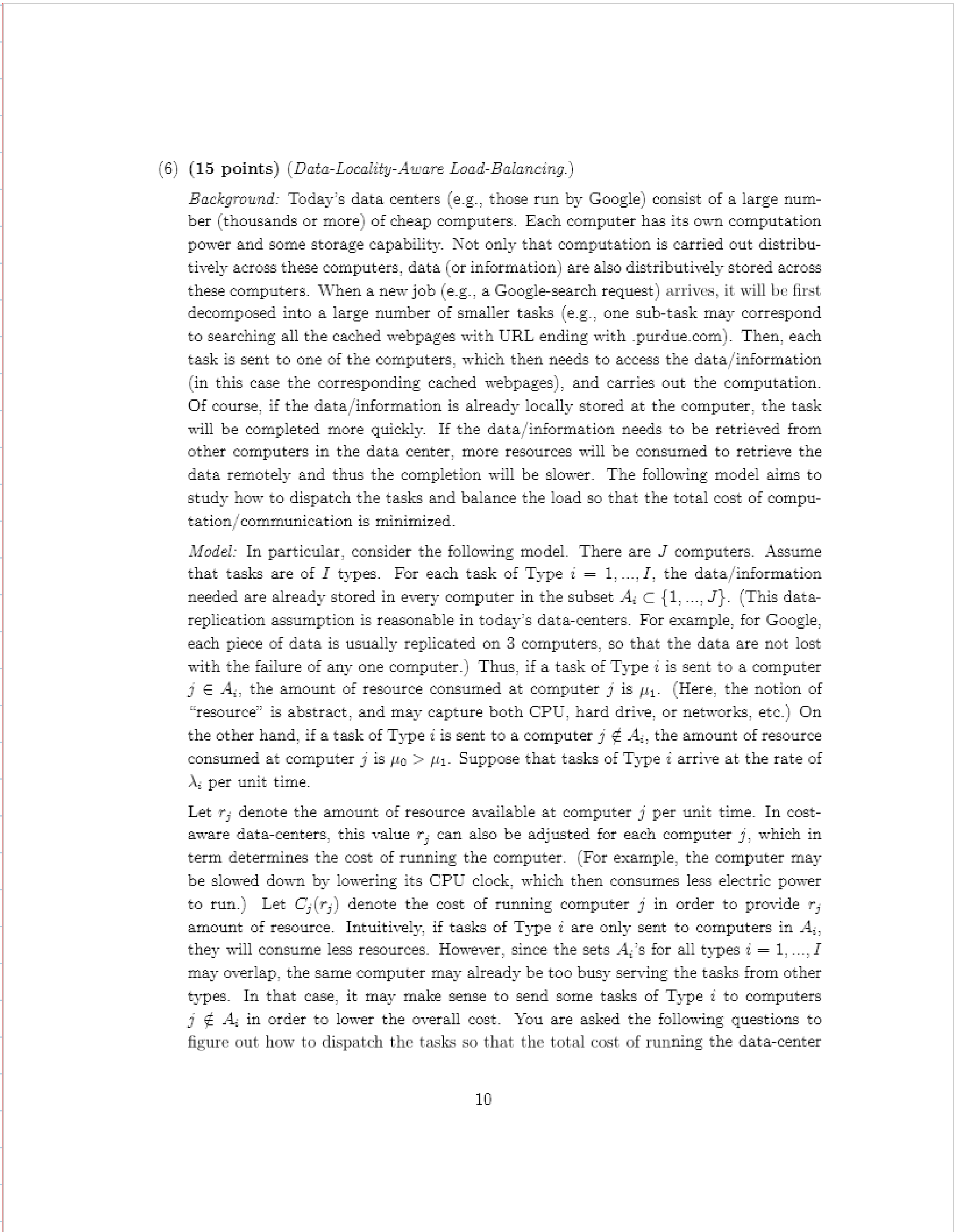
$$= \sum_{i=1}^{n} \left(w_i \left[e^{-\frac{\lambda}{w_i} - 1} \right] \left[-\frac{\lambda}{w_i} - 1 \right] + \lambda \left[e^{-\frac{\lambda}{w_i} - 1} \right] \right)$$

$$= -\sum_{i=1}^{n} w_i e^{-\frac{\lambda}{w_i} - 1} - \lambda$$

Hence, the dual problem is

$$\max \quad -\sum_{i=1}^{n} w_i e^{-\frac{\lambda}{w_i}-1} - \lambda$$

$$\text{sub to} \quad \lambda \geq 0$$



(6) **(15 points)** (*Data-Locality-Aware Load-Balancing.*)

Background: Today's data centers (e.g., those run by Google) consist of a large number (thousands or more) of cheap computers. Each computer has its own computation power and some storage capability. Not only that computation is carried out distributively across these computers, data (or information) are also distributively stored across these computers. When a new job (e.g., a Google-search request) arrives, it will be first decomposed into a large number of smaller tasks (e.g., one sub-task may correspond to searching all the cached webpages with URL ending with .purdue.com). Then, each task is sent to one of the computers, which then needs to access the data/information (in this case the corresponding cached webpages), and carries out the computation. Of course, if the data/information is already locally stored at the computer, the task will be completed more quickly. If the data/information needs to be retrieved from other computers in the data center, more resources will be consumed to retrieve the data remotely and thus the completion will be slower. The following model aims to study how to dispatch the tasks and balance the load so that the total cost of computation/communication is minimized.

Model: In particular, consider the following model. There are J computers. Assume that tasks are of I types. For each task of Type $i = 1, ..., I$, the data/information needed are already stored in every computer in the subset $A_i \subset \{1, ..., J\}$. (This data-replication assumption is reasonable in today's data-centers. For example, for Google, each piece of data is usually replicated on 3 computers, so that the data are not lost with the failure of any one computer.) Thus, if a task of Type i is sent to a computer $j \in A_i$, the amount of resource consumed at computer j is μ_1. (Here, the notion of "resource" is abstract, and may capture both CPU, hard drive, or networks, etc.) On the other hand, if a task of Type i is sent to a computer $j \notin A_i$, the amount of resource consumed at computer j is $\mu_0 > \mu_1$. Suppose that tasks of Type i arrive at the rate of λ_i per unit time.

Let r_j denote the amount of resource available at computer j per unit time. In cost-aware data-centers, this value r_j can also be adjusted for each computer j, which in term determines the cost of running the computer. (For example, the computer may be slowed down by lowering its CPU clock, which then consumes less electric power to run.) Let $C_j(r_j)$ denote the cost of running computer j in order to provide r_j amount of resource. Intuitively, if tasks of Type i are only sent to computers in A_i, they will consume less resources. However, since the sets A_i's for all types $i = 1, ..., I$ may overlap, the same computer may already be too busy serving the tasks from other types. In that case, it may make sense to send some tasks of Type i to computers $j \notin A_i$ in order to lower the overall cost. You are asked the following questions to figure out how to dispatch the tasks so that the total cost of running the data-center is minimized.

(a) (5 points) Suppose that the arrival rates λ_i's are given. Let ρ_{ij} denote the fraction of tasks of Type-i that are dispatched to computer j. Write down an optimization problem for minimizing the total cost of running the computers, subject to the constraint that the total amount of resources per unit time consumed by the tasks at each computer j is no greater than the resource available at computer j. The variables to be optimized are r_j's and ρ_{ij}'s. State the conditions under which your optimization problem will be convex.

(b) (5 points) Assume that the optimization problem is convex. Using the KKT condition, show that the optimal solution is of the following form: There exists a dual variable ν_j for each computer j such that (i) a task of Type-i will be sent to a computer $j \in A_i$ (i.e., $\rho_{ij} > 0$) only if

$$\mu_1 \underset{k\in A_i}{\overset{\min}{\cancel{\max}}} \nu_k \leq \mu_0 \underset{k\notin A_i}{\overset{\min}{\cancel{\max}}} \nu_k;$$

and (ii) a task of Type-i will be sent to a computer $j \notin A_i$ (i.e., $\rho_{ij} > 0$) only if

$$\mu_1 \underset{k\in A_i}{\overset{\min}{\cancel{\max}}} \nu_k \geq \mu_0 \underset{k\notin A_i}{\overset{\min}{\cancel{\max}}} \nu_k;$$

(c) (5 points) Using the above knowledge, write down a distributed and iterative algorithm that can be used to find the optimal primal and dual solutions to the optimization problem. (You do NOT need to prove the convergence of your algorithm.)

(a) The optimization problem is

$$\min \sum_{j=1}^{J} C_j(r_j)$$

Sub to $\displaystyle\sum_{i:j\in A_i} \lambda_i \rho_{ij} \mu_1 + \sum_{i:j\notin A_i} \lambda_i \rho_{ij} \mu_0 \leq r_j$ for all $j = 1, 2, \cdots J$

$\displaystyle\sum_{j=1}^{J} \rho_{ij} = 1$ for all $i = 1, 2, \cdots I$

(b) Associate a Lagrange multiplier ν_j for each constraint. The Lagrangian is

$$L(\vec{r}, \vec{\rho}, \vec{\nu}) = \sum_{j=1}^{J} C_j(r_j)$$

$$+ \sum_{j=1}^{J} \nu_j \left\{ \sum_{i:j\in A_i} \lambda_i \rho_{ij} \mu_1 + \sum_{i:j\notin A_i} \lambda_i \rho_{ij} \mu_0 - r_j \right\}$$

$$= \sum_{j=1}^{J} \left\{ C_j(r_j) - \nu_j r_j \right\}$$

$$+ \sum_{i=1}^{I} \lambda_i \left[\sum_{j\in A_i} \nu_j \rho_{ij} \mu_1 + \sum_{j\notin A_i} \nu_j \rho_{ij} \mu_0 \right]$$

According to the KKT condition, any primal solution should minimize the Lagrange given the dual solution.

By minimizing $\sum_{j\in A_i} \nu_j \rho_{ij} \mu_1 + \sum_{j\notin A_i} \nu_j \rho_{ij} \mu_0$

it is clear that $\rho_{ij} > 0$ only for those computers with the smallest weight, where the weight for a computer $j \in A_i$ is $\nu_j \mu_1$, and the weight for a computer $j \in A_i$ is $\nu_j \mu_0$.

The set of conditions in part (b) thus follows.

(c). The following distributed algorithm can be used:

For each Type-i: if $\min_{j \in A_i} \nu_j \mu_1 \le \min_{j \notin A_i} \nu_j \mu_0$, send the tasks to computer $j \in A_i$ with the smallest ν_j. Otherwise, send the tasks to computers $j \notin A_i$ with the smallest ν_j.

$\Rightarrow$ let $p_{ij}(t)$ be the corresponding fractions

For computer j, set $r_j(t)$ by

$$C_j'(r_j) = \nu_j$$

Finally, update ν_j by

$$\nu_j(t+1) = \left[\nu_j(t) + \alpha \left(\sum_{i: j \in A_i} \lambda_i p_{ij}(t) \mu_1 + \sum_{i: j \notin A_i} \lambda_i p_{ij}(t) \mu_0 - r_j(t) \right) \right]^+$$

(7) **(20 points)** (*LASSO: Least square with L_1-regularization.*)

Background: Suppose that an observed quantity $y \in \mathbf{R}$ is linearly dependent on other observed quantities $x_1, ..., x_p \in \mathbf{R}$. In other words, there exist coefficients $\bar{a}_1, ..., \bar{a}_p$ such that $y = \sum_{i=1}^{p} \bar{a}_i x_i$. However, we do not know $\bar{a}_1, ..., \bar{a}_p$. Rather, we can obtain n samples of these observations $[y^j, x_1^j, ..., x_p^j]$, where $j = 1, ..., n$. Thus, for each j,

$$y_j = \sum_{i=1}^{p} \bar{a}_i x_i^j.$$

We may then estimate the coefficients $\bar{a}_1, ..., \bar{a}_p$ by solving a least-square problem, i.e. by minimizing

$$\frac{1}{2} \sum_{j=1}^{n} \left[y^j - \sum_{i=1}^{p} a_i x_i^j \right]^2, \qquad (1)$$

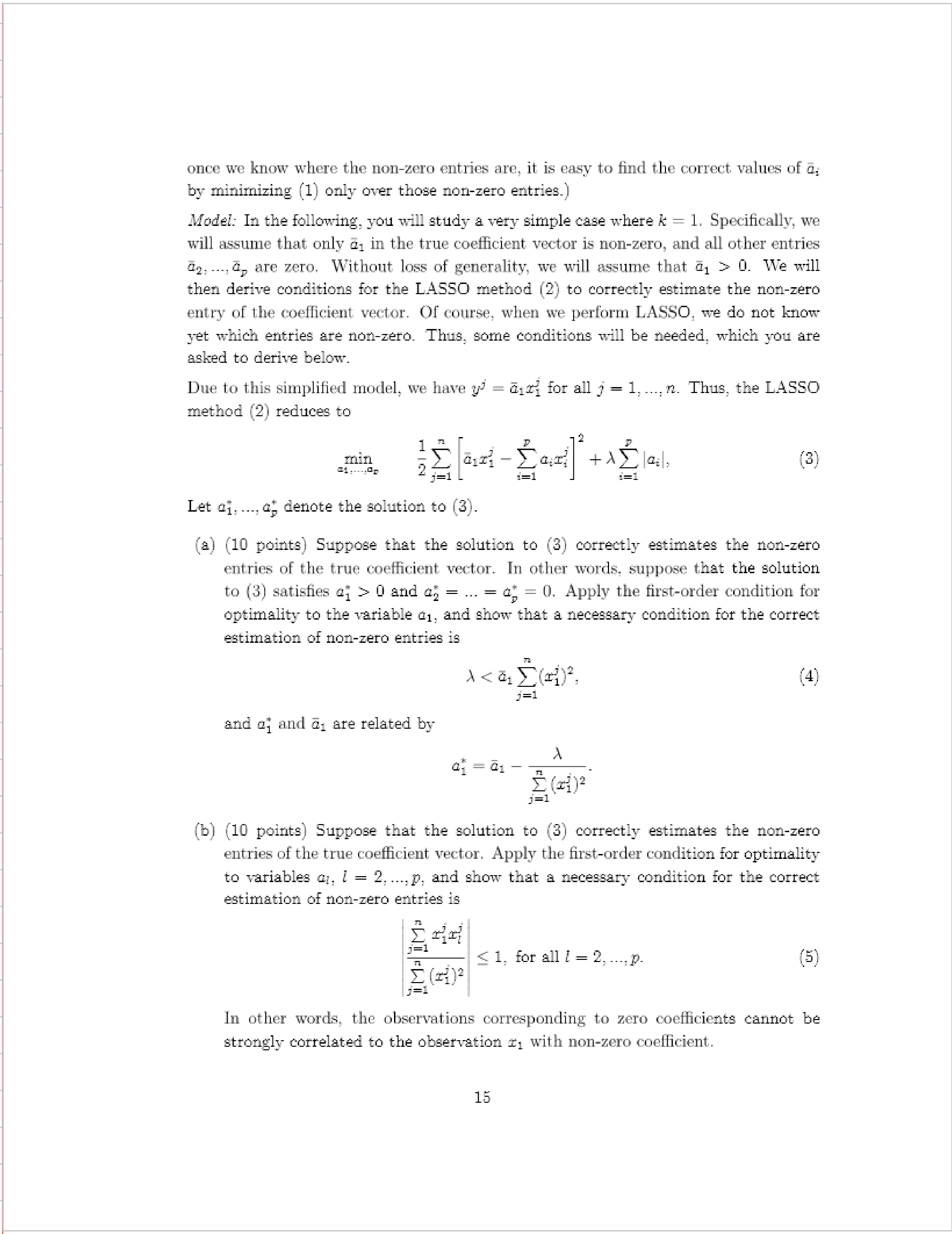
over all $a_1, ..., a_p$. Typically, if $n \geq p$ and some linear independence conditions are met, the only solution that minimizes (1) is when $a_i = \bar{a}_i$ for all i, in which case the objective function (1) will be trivially zero.

However, in the so-called *high-dimensional* problems, the dimension p may be very large, while the number of observations n may be much smaller than p. In that case, the above least-square problem will produce multiple solutions for $[a_1, ..., a_p]$ that all make the value of (1) zero. Then, it is unclear which solution represents the true coefficient vector $[\bar{a}_1, .., \bar{a}_p]$.

Fortunately, in a lot of these high-dimensional problems, the true coefficient vector $[\bar{a}_1, ..., \bar{a}_p]$ is known to be *sparse*. Specifically, for a k-sparse problem, we know in advance that only k of the coefficients $\bar{a}_1, ..., \bar{a}_p$ are non-zero. We assume that $k < n < p$. Thus, the number of samples is greater than the sparsity level, but is smaller than the total number of dimensions. Then, it makes sense to minimize (1) only over those coefficient vectors $[a_1, .., a_p]$ that meet the k-sparsity constraint. However, searching over such a space of sparse coefficient vectors is a non-convex problem. Instead, the LASSO method attempts to solve the following problem

$$\min_{a_1, ..., a_p} \quad \frac{1}{2} \sum_{j=1}^{n} \left[y^j - \sum_{i=1}^{p} a_i x_i^j \right]^2 + \lambda \sum_{i=1}^{p} |a_i|, \qquad (2)$$

where $\lambda > 0$ is an appropriately chosen constant. The hope is that adding the L_1-norm $\sum_{i=1}^{p} |a_i|$ to the minimization will force most a_i's to zero. Thus, the optimal solution $[a_1, ..., a_p]$ to (2) may correctly estimate the location of the non-zero entries in the true coefficient vector $[\bar{a}_1, ..., \bar{a}_p]$, i.e., we may have $a_i \neq 0$ if and only if $\bar{a}_i \neq 0$. (Note that the non-zero entries a_i solving (2) may still differ from the true values of $\bar{a}_i$. However,



once we know where the non-zero entries are, it is easy to find the correct values of $\bar{a}_i$ by minimizing (1) only over those non-zero entries.)

Model: In the following, you will study a very simple case where $k = 1$. Specifically, we will assume that only $\bar{a}_1$ in the true coefficient vector is non-zero, and all other entries $\bar{a}_2, ..., \bar{a}_p$ are zero. Without loss of generality, we will assume that $\bar{a}_1 > 0$. We will then derive conditions for the LASSO method (2) to correctly estimate the non-zero entry of the coefficient vector. Of course, when we perform LASSO, we do not know yet which entries are non-zero. Thus, some conditions will be needed, which you are asked to derive below.

Due to this simplified model, we have $y^j = \bar{a}_1 x_1^j$ for all $j = 1, ..., n$. Thus, the LASSO method (2) reduces to

$$\min_{a_1,...,a_p} \quad \frac{1}{2}\sum_{j=1}^{n}\left[\bar{a}_1 x_1^j - \sum_{i=1}^{p} a_i x_i^j\right]^2 + \lambda \sum_{i=1}^{p} |a_i|, \tag{3}$$

Let $a_1^*, ..., a_p^*$ denote the solution to (3).

(a) (10 points) Suppose that the solution to (3) correctly estimates the non-zero entries of the true coefficient vector. In other words, suppose that the solution to (3) satisfies $a_1^* > 0$ and $a_2^* = ... = a_p^* = 0$. Apply the first-order condition for optimality to the variable a_1, and show that a necessary condition for the correct estimation of non-zero entries is

$$\lambda < \bar{a}_1 \sum_{j=1}^{n} (x_1^j)^2, \tag{4}$$

and a_1^* and $\bar{a}_1$ are related by

$$a_1^* = \bar{a}_1 - \frac{\lambda}{\sum_{j=1}^{n} (x_1^j)^2}.$$

(b) (10 points) Suppose that the solution to (3) correctly estimates the non-zero entries of the true coefficient vector. Apply the first-order condition for optimality to variables a_l, $l = 2, ..., p$, and show that a necessary condition for the correct estimation of non-zero entries is

$$\left|\frac{\sum_{j=1}^{n} x_1^j x_l^j}{\sum_{j=1}^{n} (x_1^j)^2}\right| \leq 1, \text{ for all } l = 2, ..., p. \tag{5}$$

In other words, the observations corresponding to zero coefficients cannot be strongly correlated to the observation x_1 with non-zero coefficient.

(a) Let $f(\vec{a})$ denote the objective function

The first order condition for optimality specifies that

$$\frac{\partial f}{\partial a_1} = -\sum_{j=1}^{n} \left(\bar{a}_1 x_1^j - \sum_{i=1}^{p} a_i X_i^j \right) \cdot X_i^j + \lambda = 0$$

(Note that the derivative of $|a_1|$ is 1 when $a_1 > 0$)

Substituting $a_2^* = \cdots = a_p^* = 0$,

we have

$$-\sum_{j=1}^{n} (\bar{a}_1 - a_1^*) (X_1^j)^2 + \lambda = 0$$

Thus, in order to have $a_1 > 0$, we must have

$$\lambda = \sum_{j=1}^{n} (\bar{a}_1 - a_1^*)(X_1^j)^2 < \bar{a}_1 \sum_{j=1}^{n} (X_1^j)^2$$

and

$$a_1^* = \bar{a}_1 - \frac{\lambda}{\sum_{j=1}^{n} (X_1^j)^2}$$

(b) Apply the first order condition to variable a_l, $l = 2, \dots, p$. Note that at $a_l = 0$, the derivative of $|a_l|$ does not exist. However, we know that the sub-gradient must be within $[-1, 1]$

Hence, there must be a number $\nu \in [-1, 1]$ such that

$$\frac{\partial f}{\partial a_l} = -\sum_{j=1}^{n} \left(\bar{a}_1 x_1^j - \sum_{i=1}^{p} a_i x_i^j \right) x_l^j + \lambda \cdot \nu = 0$$

Substituting $a_1^* = \bar{a}_1 - \frac{\lambda}{\sum_{j=1}^{n} (x_1^j)^2}$

$a_2^* = \cdots a_p^* = 0$, we obtain

$$\sum_{j=1}^{n} \frac{\lambda}{\sum_{j=1}^{n} (x_1^j)^2} \cdot x_1^j x_l^j + \lambda \nu = 0$$

$$\Rightarrow \lambda \frac{\sum_{j=1}^{n} x_1^j x_l^j}{\sum_{j=1}^{n} (x_1^j)^2} + \lambda \nu = 0$$

Since $\nu \in [-1, 1]$, we must have

$$\left| \frac{\sum_{j=1}^{n} x_1^j x_l^j}{\sum_{j=1}^{n} (x_1^j)^2} \right| \le 1$$