Convex Optimization: Homework 4

Due to Nov. 26, 2023

Problem 1: Derive a dual problem for QCQP.

Problem 2: Derive a dual problem for

$$\text{minimize} \qquad \sum_{i=1}^{N} \|\mathbf{A}_i\mathbf{x} + \mathbf{b}_i\|_2 + \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2.$$

The problem data are $\mathbf{A}_i \in R^{m_i \times n}$, $\mathbf{b}_i \in R^{m_i}$, and $\mathbf{x}_0 \in R^n$. First introduce new variables $\mathbf{y}_i \in R^{m_i}$ and equality constraints $\mathbf{y}_i = \mathbf{A}_i\mathbf{x} + \mathbf{b}_i$.

Problem 3: *Lagrangian relaxation of Boolean LP.*
A Boolean linear program is an optimization problem of the form

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \preceq \mathbf{b}, \qquad x_i \in \{0,1\}, \quad i = 1, \cdots, n. \end{array}$$

and is, in general, very difficult to solve. In exercise 4.15 of the Textbook, the following LP relaxation of this problem,

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \preceq \mathbf{b}, \qquad 0 \le x_i \le 1, \quad i = 1, \cdots, n. \end{array}$$

is considered. It is much easier to solve, and gives a lower bound on the optimal value of the Boolean LP. However, we can derive another lower bound for the Boolean LP using Duality, and work out the relation between the two lower bounds (one in exercise 4.15 of the Textbook and the other that you derive through Duality).

- *Lagrangian relaxation method:* The Boolean LP can be reformulated as the following problem

$$\begin{array}{ll} \text{minimize} & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & \mathbf{A}\mathbf{x} \preceq \mathbf{b}, \qquad x_i(1 - x_i) = 0, \quad i = 1, \cdots, n. \end{array}$$

which has quadratic equality constraints. Find the Lagrange dual of this problem. The optimal value of the dual problem (which is convex) gives a lower bound on the optimal value of the Boolean LP. This method of finding a lower bound on the optimal value is called Lagrangian relaxation.

- Show that the lower bound obtained via Lagrangian relaxation, and via the LP relaxation in exercise 4.15 of the Textbook, are the same. *Hint.* Derive the dual of the LP relaxation (5.107) in exercise 4.15 of the Textbook.

Problem 4: *Equality constrained least-squares.*
Consider the equality constrained least-squares problem

$$\begin{array}{ll} \text{minimize} & \|\mathbf{A}\mathbf{x} + b\|_2^2 \\ \text{subject to} & \mathbf{G}\mathbf{x} = \mathbf{h} \end{array}$$

where $\mathbf{A} \in R^{m\times n}$ with $\operatorname{rank}\mathbf{A} = n$, and $\mathbf{G} \in R^{p\times n}$ with $\operatorname{rank}\mathbf{G} = p$.

Give the KKT conditions, and derive expressions for the primal solution $\mathbf{x}^*$ and the dual solution ν^*.

Problem 5: *Estimation of covariance and mean of a multivariate normal distribution.*

Consider the problem of estimating the covariance matrix $\mathbf{R}$ and the mean $\mathbf{a}$ of a Gaussian probability density function

$$p_{\mathbf{R},\mathbf{a}} = (2\pi)^{-\frac{1}{2}} (\det\{\mathbf{R}\})^{-\frac{n}{2}} \exp\{-(\mathbf{y}-\mathbf{a})^T\mathbf{R}^{-1}(\mathbf{y}-\mathbf{a})\},$$

based on N independent samples $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N \in R^n$.

- First consider the estimation problem when there are no additional constraints on $\mathbf{R}$ and $\mathbf{a}$. Let μ and $\mathbf{Y}$ be the sample mean and covariance, defined as

  $$\mu = \frac{1}{N}\sum_{k=1}^{N}\mathbf{y}_k, \qquad \mathbf{Y} = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{y}_k - \mu)(\mathbf{y}_k - \mu)^T$$

  Show that the log-likelihood function

  $$l(\mathbf{R},\mathbf{a}) = -\frac{Nn}{2}\log(2\pi) - \frac{N}{2}\log\det\{\mathbf{R}\} - \frac{1}{2}\sum_{k=1}^{N}(\mathbf{y}_k - \mathbf{a})^T\mathbf{R}^{-1}(\mathbf{y}_k - \mathbf{a})$$

can be expressed as

$$l(\mathbf{R}, \mathbf{a}) = \frac{N}{2}\Big(-n\log(2\pi) - \log\det\{\mathbf{R}\} - \mathrm{tr}(\mathbf{R}^{-1}\mathbf{Y}) - (\mathbf{a}-\mu)^T\mathbf{R}^{-1}(\mathbf{a}-\mu)\Big).$$

Use this expression to show that if $\mathbf{Y} \succ \mathbf{0}$, the ML estimates of $\mathbf{R}$ and $\mathbf{a}$ are unique, and given by

$$\mathbf{a}_{\mathrm{ML}} = \mu, \qquad \mathbf{R}_{\mathrm{ML}} = \mathbf{Y}.$$

- The log-likelihood function includes a convex term $-\log\det\{\mathbf{R}\}$, so it is not obviously concave. Show that $l(\mathbf{R}, \mathbf{a})$ is concave, jointly in $\mathbf{R}$ and $\mathbf{a}$, in the region defined by

$$\mathbf{R} \preceq 2\mathbf{Y}.$$

This means we can use convex optimization to compute simultaneous ML estimates of $\mathbf{R}$ and $\mathbf{a}$, subject to convex constraints, as long as the constraints include $\mathbf{R} \preceq 2\mathbf{Y}$, i.e., the estimate $\mathbf{R}$ must not exceed twice the unconstrained ML estimate.

Problem 6: *Barrier method for second-order cone programming.*
Consider the SOCP (without equality constraints, for simplicity)

$$\begin{array}{ll} \text{minimize} & \mathbf{f}^T\mathbf{x} \\ \text{subject to} & \|\mathbf{A}_i\mathbf{x} + b_i\|_2 \le \mathbf{c}_i^T\mathbf{x} + d_i, \qquad i = 1, \ldots, m. \end{array} \tag{1}$$

The constraint functions in this problem are not differentiable (since the Euclidean norm $\|\mathbf{u}\|_2$ is not differentiable at $\mathbf{u} = 0$) so the (standard) barrier method cannot be applied. In the Textbook (see Example 11.8, page 599, and page 601.), it is shown that this SOCP can be solved by an extension of the barrier method that handles generalized inequalities.

In this problem, you need to show how the standard barrier method (with scalar constraint functions) can be used to solve the SOCP.

We first reformulate the SOCP as

$$\begin{array}{ll} \text{minimize} & \mathbf{f}^T\mathbf{x} \\ \text{subject to} & \dfrac{\|\mathbf{A}_i\mathbf{x} + b_i\|_2^2}{\mathbf{c}_i^T\mathbf{x} + d_i} \le \mathbf{c}_i^T\mathbf{x} + d_i, \qquad i = 1, \ldots, m \\ & \mathbf{c}_i^T\mathbf{x} + d_i > 0, \qquad i = 1, \ldots, m. \end{array} \tag{2}$$

The constraint function

$$f_i(x) = \frac{\|\mathbf{A}_i\mathbf{x} + b_i\|_2^2}{\mathbf{c}_i^T\mathbf{x} + d_i} - \mathbf{c}_i^T\mathbf{x} - d_i$$

is the composition of a quadratic-over-linear function with an affine function, and is twice differentiable (and convex), provided we define its domain as $\mathbf{dom} f_i = \{\mathbf{x} \mid \mathbf{c}^T\mathbf{x}_i + d_i > 0\}$. Note that the two problems (1) and (2) are not exactly equivalent. If $\mathbf{c}_i^T\mathbf{x}^* + d_i = 0$ for some i, where $\mathbf{x}^*$ is the optimal solution of the SOCP (1), then the reformulated problem (2) is not solvable; $\mathbf{x}$ is not in its domain. Nevertheless we will see that the barrier method, applied to (2), produces arbitrarily accurate suboptimal solutions of (2), and hence also for (1).

1. Form the log barrier ϕ for the problem (2). Compare it to the log barrier that arises when the SOCP (1) is solved using the barrier method for generalized inequalities.

2. Show that if $t\mathbf{f}^T\mathbf{x} + \phi(\mathbf{x})$ is minimized, the minimizer $\mathbf{x}^*(t)$ is $2m/t$-suboptimal for the problem (1). It follows that the standard barrier method, applied to the reformulated problem (2), solves the SOCP (1), in the sense of producing arbitrarily accurate suboptimal solutions. This is the case even though the optimal point $\mathbf{x}^*$ need not be in the domain of the reformulated problem (2).

Problem 7: Implement **Gradient Descent** and **Newton** algorithms for the problem

$$\begin{aligned} \text{minimize}_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} - \sum_{i=1}^{m} \log(\mathbf{a}_i^T\mathbf{x} + b_i) \\ \text{such that} \quad & \mathbf{a}_i^T\mathbf{x} + b_i > 0, \quad i = 1, \cdots, m \end{aligned}$$

$m = 100$, $n = 50$. See lec07.pdf: slide 8 for the problem, slide 6 for **Gradient Descent** algorithm, and slide 14 for **Newton** algorithm. Generate vectors $\mathbf{c}$, $\mathbf{a}_i$, $i = 1, \cdots, m$, and values b_i arbitrarily, but so that the problem would make sense. Reproduce and report the results shown in slides 8 and 16 lec07.pdf. As a report submit your codes (preferably in Matlab or Python) and figures. If the time is limited, implement at least Gradient Descent algorithm with fixed step size.

Hint: To avoid issues with random number generator in Matlab when creating the LP: $\min\{\mathbf{c}^T\mathbf{x} \mid \mathbf{a}_i^T\mathbf{x} + b_i > 0, \quad i = 1, \cdots, m\}$, you can use the attached simple Matlab script for generating a valid LP instance. Feel free to include the code as a validation stage in the submitted code. This is entirely optional.