UC Berkeley
Department of Electrical Engineering and Computer Science

EECS 227A
Nonlinear and Convex Optimization

Solutions 6
Fall 2009

Solution 6.1

(a) $p^* = 1$

(b) The Lagrangian is $L(x, y, \lambda) = e^{-x} + \lambda x^2/y$. The dual function is

$$g(\lambda) = \inf_{x,y>0}(e^{-x} + \lambda x^2/y) = \begin{cases} 0 & \text{if } \lambda \geq 0 \\ -\infty & \text{otherwise} \end{cases}$$

so we can write the dual problem as

$$\begin{array}{ll} \text{maximize} & 0 \\ \text{subject to} & \lambda \geq 0 \end{array}$$

with optimal value $d^* = 0$. The optimal duality gap is $p^* - d^* = 1$

(c) Slater's condition is not satisfied.

(d) $p^*(u) = 1$ if $u = 0, p^*(u) = 0$ if $u > 0$ and $p^*(u) = \infty$ if $u < 0$

Solution 6.2

Suppose x is feasible. Since f_i are convex and $f_i(x) \leq 0$, we have

$$0 \geq f_i(x) \geq f_i(x^*) + \nabla f_i(x^*)^T(x - x^*), i = 1, \ldots, m$$

Using $\lambda_i^* \geq 0$, we conclude that

$$\begin{aligned} 0 &\geq \sum_{i=1}^m \lambda_i^*(f_i(x^*) + \nabla f_i(x^*)^T(x - x^*) \\ &= \sum_{i=1}^m \lambda_i^*(f_i(x^*) + \sum_{i=1}^m \nabla f_i(x^*)^T(x - x^*) \\ &= -\nabla f_0(x^*)^T(x - x^*) \end{aligned}$$

In the last line, we use the complementary slackness condition $\lambda_i^* f_i(x^*) = 0$, and the last KKT condition. This show that $\nabla f_0(x^*)^T(x - x^*) \geq 0$, i.e. $\nabla f_0(x^*)$ defines a supporting hyperplane to feasible set at x^*

Solution 6.3

(a) Follows from $\mathrm{tr}(Wxx^T) = x^TWx$ and $(xx^T)_{ii} = x_i^2$

(b) It gives a lower bound because we minimize the same objective function over a larger set. If X is rank one, it is optimal.

(c) We write the problem as a minimization problem

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T \nu \\ \text{subject to} & W + \mathbf{diag}(\nu) \succeq 0 \end{array}$$

Introducing a Lagrange multiplier $X \in \mathbf{S}^n$ for the matrix inequality, we obtain the Lagrangian

$$\begin{aligned} L(\nu, X) &= \mathbf{1}^T \nu - \operatorname{tr}(X(W + \mathbf{diag}(\nu))) \\ &= \mathbf{1}^T \nu - \operatorname{tr}(XW) - \textstyle\sum_{i=1}^n \nu_i X_{ii} \\ &= -\operatorname{tr}(XW) + \textstyle\sum_{i=1}^n \nu_i (1 - X_{ii}) \end{aligned}$$

This is bounded below as a function of ν only if $X_{ii} = 1$ for all i, so we obtain the dual problem

$$\begin{array}{ll} \text{maximize} & -\operatorname{tr}(WX) \\ \text{subject to} & X \succeq 0 \\ & X_{ii} = 1, i = 1, \ldots, n \end{array}$$

Changing the sign again, and switching from maximization to minimization, yields the problem in part (a)

Solution 6.4

(a) We introduce the new variables, and write the problem as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \|y_i\|_2 \leq t_i, i = 1, \ldots, m \\ & y_i = A_i x + b_i, i = 1, \ldots, m \\ & t_i = c_i^T x + d_i, i = 1, \ldots, m \end{array}$$

The Lagrangian is

$$\begin{aligned} L(x, y, t, \lambda, \nu, \mu) &= c^T x + \sum_{i=1}^m \lambda_i (\|y_i\|_2 - t_i) + \sum_{i=1}^m \nu_i^T (y_i - A_i x - b_i) \\ &+ \sum_{i=1}^m \mu_i (t_i - c_i^T x - d_i) \\ &= (c - \sum_{i=1}^m A_i^T \nu_i - \sum_{i=1}^m \mu_i c_i)^T x + \sum_{i=1}^m (\lambda_i \|y_i\|_2 + \nu_i^T y_i) \\ &+ \sum_{i=1}^m (-\lambda_i + \mu_i) t_i - \sum_{i=1}^m (b_i^T \nu_i + d_i \mu_i) \end{aligned}$$

The minimum over x is bounded below if and only i

$$\sum_{i=1}^m (A_i^T \nu_i + \mu_i c_i) = c$$

To minimize over y_i, we note that

$$\inf_{y_i} (\lambda_i \|y\|_i + \nu_i^T y_i) = \begin{cases} 0 & \|\nu_i\|_2 \leq \lambda_i \\ -\infty & \text{otherwise} \end{cases}$$

The minimum over t_i is bounded below if and only if $\lambda_i = \mu_i$. The Lagrangian is

$$g(\lambda, \nu, \mu) = \begin{cases} -\sum_{i=1}^m (b_i^T \nu_i + d_i \mu_i) & \sum_{i=1}^m (A_i^T \nu_i + \mu_i c_i) = c, \|\nu_i\|_2 \le \lambda_i, \mu = \lambda \\ -\infty & \text{otherwise} \end{cases}$$

which leads to the dual problem

$$\begin{array}{ll} \text{maximize} & -\sum_{i=1}^m (b_i^T \nu_i + d_i \lambda_i) \\ \text{subject to} & \sum_{i=1}^m (A_i^T \nu_i + \lambda_i c_i) = c \\ & \|\nu_i\|_2 \le \lambda_i, i = 1, \ldots, m \end{array}$$

(b) We express the SOCP as a conic form problem

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & -(A_i x + b_i, c_i^T x + d_i) \preceq_{K_i} 0, i = 1, \ldots, m \end{array}$$

The conic dual is

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^m (b_i^T u_i + d_i v_i) \\ \text{subject to} & \sum_{i=1}^m (A_i^T u_i + v_i c_i) = c \\ & (u_i, v_i) \succeq_{K_i^*} 0, i = 1, \ldots, m \end{array}$$

Solution 6.5

(a) Since f is a convex and closed function (i.e., its epigraph is a closed set), it can be represented via its conjugate, as

$$f(r) = \max_y \left\{ r^T y - f^*(y) \right\}.$$

Consequently, we can express the problem in minimax form, as $p^* = \min_x \max_y \phi(x, y)$, where the function

$$\phi(x, y) := y^T (Ax + b) - f^*(y) + \frac{1}{2}\|x\|_2^2.$$

Weak duality tells us that $p^* \ge d^*$, where

$$d^* := \max_y \min_x \phi(x, y).$$

We obtain

$$-d^* = \min_y \left\{ f^*(y) + \frac{1}{2}\|A^T y\|_2^2 - b^T y \right\}.$$

(b) We observe that for every y, the sub-level sets of the function $\phi(\cdot, y)$ are bounded. (Here to avoid trivial sub-cases, we assume that p^* is finite, a condition that should have been in the problem statement.) Thus, according to the result of [BV,exercise 5.25], we have $p^* = Ad^*$. We observe that d^* (hence, p^*) is convex in $K = AA^T$, since $-d^*$ is concave:

$$-d^* = \min_y \left\{ f^*(y) + \frac{1}{2} y^T K y - b^T y \right\}.$$

(c) The primal problem involves a strictly convex objective function and no constraints, hence the optimum is attained and unique. For each y, the problem

$$\min_x \phi(x, y)$$

has a unique solution, given by $x(y) := A^T y$. According to the result in [BV,§5.5.5], we conclude that if y^* is optimal for the dual problem, then $x^* = A^T y^*$ is optimal.

(d) The dual takes the following specific forms.

(i) *Support vector machines classification:* When $f(r) = \sum_{i=1}^m (r_i)_+$, we have

$$f(r) = \max_{0 \le u \le \mathbf{1}} u^T r,$$

which shows that f^* is then the indicator function of the set $[0, 1]^m$. The dual problem writes (with $b = \mathbf{1}$):

$$-2d^* = \min_y \|A^T y\|_2^2 - 2b^T y \ : \ 0 \le y \le \mathbf{1}.$$

(ii) *Least-squares regression:* The function $f(r) = \frac{1}{2}\|r\|_2^2$ is self-conjugate, so that the dual problem takes the form

$$2d^* = \min_y \frac{1}{2} 2b^T y - y^T (K + I) y = b^T (K + I)^{-1} b,$$

as expected from the primal form.

(ii) *Least-norm regression:* when $f(r) = \|r\|$, where $\|\cdot\|$ is a norm, the conjugate of f is the indicator of the unit ball for the dual norm, hence

$$-2d^* = \min_y \|A^T y\|_2^2 - 2b^T y \ : \ \|y\|_* \le 1.$$

We can express d^* as

$$-d^* = \min_y \|y - y_0\|_K \ : \ \|y\|_* \le 1,$$

where $y_0 := K^{-1} b$, and $\|\cdot\|_K$ is the weighted Euclidean norm with values $\|z\|_K^2 = z^T K z$. The above represented the minimum weighted Euclidean distance from y_0 the unit ball in the dual norm.

Solution 6.6

(a) The KKT conditions for (z^*, λ^*) are given by the following equations:

$$\nabla_z L(z^*, \lambda^*) = \nabla f(x^n) + \nabla^2 f(x^n) z^* + A^T \lambda^* = 0 \quad (1a)$$

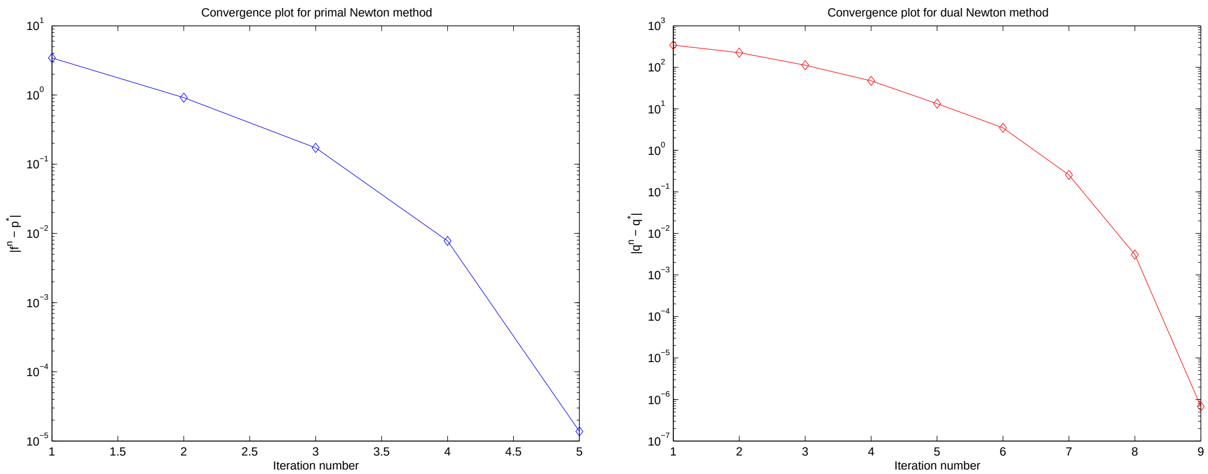
$$A z^* = 0. \quad (1b)$$

Putting these equations together yields the given matrix form. Since the problem is strictly convex (assuming that $\nabla^2 f(x^n) \succ 0$) with linear constraints, these KKT conditions are necessary and sufficient to yield the optimum. Hence when $P(x^n)$ is invertible, solving the system yields the given form of the Newton update.

(b) Here is some MATLAB code to solve this problem via Newton's method with Armijo rule:

```
% Newton's method with Armijo rule to solve the constrained maximum
% entropy problem in primal form

clear f;

MAXITS = 500;  % Maximum number of iterations
BETA = 0.5;  % Armijo parameter
SIGMA = 0.1; % Armijo parameter

GRADTOL = 1e-7;  % Tolerance for gradient

load xinit.ascii;
load A.ascii;
load b.ascii

x = xinit;

m = size(A,1);
n = size(A,2);

for iter=1:MAXITS,
  val = x'*log(x);  % current function value
  f(iter) = val;
  grad = 1 + log(x);  % current gradient
  hess = diag(1./x);
  temp = -[hess A'; A zeros(m,m)] \ [grad; zeros(m,1)];
  newt = temp(1:n);
  primal_lambda = temp(n+1:(n+m));

  descmag = grad'*newt;  % Check magnitude of descent
  if (abs(descmag) < GRADTOL) break; end;

  t = 1;
  while (min(x + t*newt) <= 0) t = BETA*t; end;
  while ( ((x+t*newt)'*log(x+t*newt)) - val >= SIGMA*t*descmag)
       t = BETA*t;
  end;
  x = x + t*newt;
end;

gradviol = norm(A*x -b,2)
pstar = val
```

Applying to the problem data on the website yields $p^* = -33.6429$. Figure 1(a) shows $\log|f(x^n) - p^*|$ versus iteration number n.



(a) Primal Newton method

(b) Dual Newton method

Figure 1: Convergence plots for the primal Newton method (a) and dual Newton method (b). The inverted quadratic shape (on a log scale) reveals the quadratic convergence.

(c) The Lagrangian dual is given by

$$\begin{aligned} q(\lambda) \quad &:= \quad \inf_{x \succ 0} \Big\{ \sum_{i=1}^n x_i \log x_i + \lambda^T (Ax - b) \Big\} \\ &= \quad \sum_{i=1}^n \inf_{x_i > 0} \big\{ x_i \log x_i - (a_i^T \lambda) x_i \big\} - b^T \lambda \\ &= \quad - \sum_{i=1}^n g^*(-a_i^T \lambda) - b^T \lambda \end{aligned}$$

where $g(v) = v \log v$ with $\text{dom}(g) = \{v > 0\}$, and g^* is its conjugate dual. Straightforward calculations give $g^*(\mu) = \exp(\mu - 1)$ with $\text{dom}(g^*) = \mathbb{R}$, so that the result follows.

(d) Here is some MATLAB code to solve the dual problem:

```
% MATLAB code to perform dual optimization via Newton's method
% This code actually solves the convex problem of minimizing the
% negative dual function.

clear q;

MAXITS = 500;  % Maximum number of iterations
BETA = 0.5;  % Armijo parameter
SIGMA = 0.1; % Armijo parameter
```

```
GRADTOL = 1e-7;  % Tolerance for gradient

load xinit.ascii;
load A.ascii;
load b.ascii

m = size(A,1);
n = size(A,2);

laminit = ones(m,1);
lambda = laminit;

for iter=1:MAXITS,
  val = b'*lambda + sum(exp(-A'*lambda -1));  % current function value
  q(iter) = -val;
  grad = b - A*exp(-A'*lambda-1);  % current gradient
  hess = A*diag(exp(-A'*lambda-1))*A';

  newt = -hess \grad;
  descmag = grad'*newt;  % Check magnitude of descent
  if (abs(descmag) < GRADTOL) break; end;

  t = 1;
  newval = b'*(lambda + t*newt) + sum(exp(-A'*(lambda + t*newt)-1));
  while  (newval  > val + SIGMA*t*descmag)
      t = BETA*t;
      fnew = b'*(lambda + t*newt) + sum(exp(-A'*(lambda + t*newt)-1));
  end;
  lambda = lambda + t*newt;
end;

qstar = -val
```

It yields the optimal dual value $q^* = -33.6429 = p^*$, so that strong duality holds. Figure 1(b) shows a plot of $\log |q(\lambda^n) - q^*|$ versus iteration number n.