Solutions to hw8 homework on Convex Optimization
https://web.stanford.edu/class/ee364a/homework.html

Andrei Keino

November 11, 2020

9.8

Steepest descent method in l_∞-norm. Explain how to find a steepest descent direction in the l_∞-norm, and give a simple interpretation.

Solution.

The normalized steepest descent direction is given by

$$\Delta x_{nsd} = -sign(\nabla f(x))$$

where the sign is taken componentwise. Interpretation: If the partial derivative with respect to x_k is positive we take a step that reduces x_k; if it is positive, we take a step that increases x_k.

The unnormalized steepest descent direction is given by

$$\Delta x_{sd} = -||f(x)||_1 sign(\nabla f(x))$$

10.1

Nonsingularity of the KKT matrix. Consider the KKT matrix

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix}$$

where $P \in S_+^n$, $A \in R^{p \times n}$, and rank $A = p < n$.

(a) Show that each of the following statements is equivalent to nonsingularity of the KKT matrix.

1. $\mathcal{N}(P) \cap \mathcal{N}(A) = \{0\}$.

2. $Ax = 0$, $x \neq 0 \Rightarrow x^T P x \neq 0$.

3. $F^T P F \succ 0$ where $F \in R^{n \times (n-p)}$ is a matrix for which $\mathcal{R}(F) = \mathcal{N}(A)$.

4. $P + A^T Q A \succ 0$ for some $Q \succeq 0$.

Solution.

a) • *Conditions 1 and 2.* If $x \in \mathcal{N}(P) \cap \mathcal{N}(A)$, $x \neq 0$, then $Ax = 0$, $x \neq 0$ but $x^T P x = 0$, contradicting the second statement. Conversely, suppose the second statement fails to hold, *i.e.*, there is an x with $Ax = 0$, $x \neq 0$, but $x^T P x = 0$. Since $P \succeq 0$ we conclude $Px = 0$, *i.e.*, $x \in \mathcal{N}(P)$, which contradicts the first statement.

• *Conditions 2 and 3.* If $Ax = 0$, $x \neq 0$, then x must have the form $x = Fz$ where $z \neq 0$, because $rank(F) = n - p$. Then we have $x^T P x = z^T F^T P F z > 0$.

• *Conditions 2 and 4.* If the second condition holds then

$$x^T (P + A^T A) x = x^T P x + ||Ax||_2^2 > 0$$

for all nonzero x, so the last statement holds with $Q = I$. It the last statement holds for some $Q \succeq 0$ then

$$x^T (P + A^T Q A) x = x^T P x + x^T A^T Q A x > 0$$

for all nonzero x. Therefore if $Ax = 0$ and $x \neq 0$ we must have $x^T P x > 0$.

Now let us show that the four statements are equivalent to nonsingularity of the KKT matrix. First suppose that x satisfies $Ax = 0$, $Px = 0$ and $x \neq 0$. Then

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ 0 \end{bmatrix} = 0$$

which shows that the KKT matrix is singular.

Now suppose the KKT matrix is singular, i.e., there are x, z, not both zero, such that

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = 0$$

This means that $Px + A^T z = 0$ and $Ax = 0$, so multiplying the first equation by x^T on the left we get $x^T P x + x^T A^T z = 0$. Using $Ax = 0$, we get $x^T P x = 0$, so we have $Px = 0$ (using $P \succeq 0$). This contradicts (a) unless we have $x = 0$. In this case we must have $z \neq 0$. But then $A^T z = 0$ contradicts $rank(A) = p$.

b) From part (a) $P + A^T A \succ 0$. Therefore there exists a nonsingular matrix $R \in R^{n \times n}$ such that
$$R^T (P + A^T A) R = I$$

2

Let $AR = U\Sigma V_1^T$ be the singular value decomposition of AR, with $U \in R^{p \times p}$, $\Sigma = diag(\sigma_1, \ldots, \sigma_p) \in R^{p \times p}$ and $V_1 \in Rn \times p$. Let $V_2 \in R^{n \times (n-p)}$ be such that

$$V = [V_1 \quad V_2]$$

is orthogonal, and define

$$S = [\Sigma \quad 0] \in R^{p \times n}.$$

We have $AR = U\Sigma V^T$, so

$$V^T R^T (P + A^T A)RV = V^T R^T PRV + S^T S = I$$

Therefore $V^T R^T PRV = I - S^T S$ is diagonal. We denote this matrix by Λ :

$$\Lambda = V^T R^T PRV = \mathbf{diag}(1 - \sigma_1^2, \ldots, 1 - \sigma_p^2, 1, \ldots, 1).$$

applying the congruence transformation to the KKT matrix gives

$$\begin{bmatrix} V^T R^T & 0 \\ 0 & U^T \end{bmatrix} \begin{bmatrix} P & A^T \\ 0 & A \end{bmatrix} \begin{bmatrix} VR & 0 \\ 0 & U \end{bmatrix} = \begin{bmatrix} \Lambda & S^T \\ S^T & 0 \end{bmatrix}$$

and the inertia of the KKT matrix is equal to the inertia of the matrix on the right. Applying a permutation to the matrix on the right gives a block diagonal matrix with n diagonal blocks

$$\begin{bmatrix} \lambda_i & \sigma_i \\ \sigma_i & 0 \end{bmatrix}, \quad i = 1, \ldots, p \quad \lambda_i = 1, \quad i = p+1, \ldots, n.$$

The eigenvalues of the 2×2 -blocks are:

$$\frac{\lambda_i \pm \sqrt{\lambda_i^2 + 4\sigma_i^2}}{2}$$

i.e., one eigenvalue is positive and one is negative. We conclude that there are $p + (n - p) = n$ positive eigenvalues and p negative eigenvalues.

11.13

Self-concordance and negative entropy.

(a) Show that the negative entropy function $xlog(x)$ (on R_{++}) is not self-concordant.

(b) Show that for any $t > 0$, $txlog(x) - log(x)$ is self-concordant (on R_{++}).

Solution

(a) Consider the function $f(x) = x log(x)$.

$$f'(x) = log(x) + 1; \quad f''(x) = 1/x; \quad f'''(x) = -1/x^2$$

Thus

$$\frac{|f'''(x)|}{(f''(x)^{3/2})} = 1/x^{1/2}$$

and this function is not bounded on R_{++}. So, the negative entropy function is not self-concordant.

(b) Consider the function $g(x) = tx log(x) - log(x)$.

$$g'(x) = -1/x + t + t log(x) \quad g''(x) = 1/x^2 + t/x \quad g'''(x) = -2/x^3 - t/x^2$$

Then

$$\psi(x) = \frac{|f'''(x)|}{(f''(x)^{3/2})} = \frac{2/x^3 + t/x^2}{(1/x^2 + t/x)^{3/2}} = \frac{2 + tx}{(1 + tx)^{3/2}}$$

We have:

$$\psi(0) = 2, \quad \psi(\infty) = 1$$

and

$$\frac{d\psi}{d(tx)} = \frac{(1 + tx)^{3/2} - 3/2(1 + tx)^{1/2}(2 + tx)}{(1 + tx)^3} = -\frac{2 + tx/2}{(1 + tx)^{5/2}} < 0 \; \forall \, t > 0, \; x > 0$$

It means that function $\psi(tx)$ monotonically decreases on R_{++} from 2 to 1, for any $t > 0$, so we are done.

Additional exercises

In the following three exercises, you will implement a barrier method for solving the standard form LP

$$\begin{aligned}
\text{minimize} \quad & c^T x \\
\text{subject to} \quad & Ax = b, \quad x \succeq 0,
\end{aligned}$$

with variable $x \in R^n$, where $A \in R^{m \; timesn}$, with $m < n$. Throughout this exercise we will assume that A is full rank, and the sublevel sets $\{x| \; Ax = b, \; x \succeq 0 \; c^T x < \gamma\}$ are all bounded. (If this is not the case, the centering problem is unbounded below.)

1. Centering step. Implement Newton's method for solving the centering problem

$$\begin{aligned}
\text{minimize} \quad & c^T x - \sum_{i=1}^{n} log(x_i) \\
\text{subject to} \quad & Ax = b,
\end{aligned}$$

with variable x, given a strictly feasible starting point x_0.

Your code should accept A, b, c, and x_0, and return x^*, the primal optimal point, ν^*, a dual optimal point, and the number of Newton steps executed.

Use the block elimination method to compute the Newton step. (You can also compute the Newton step via the KKT system, and compare the result to the Newton step computed via block elimination. The two steps should be close, but if any x_i is very small, you might get a warning about the condition number of the KKT matrix.)

Plot $\lambda^2/2$ versus iteration k, for various problem data and initial points, to verify that your implementation gives asymptotic quadratic convergence. As stopping criterion, you can use $\lambda^2/2 < 10^{-6}$. Experiment with varying the algorithm parameters α and $beta$, observing the effect on the total number of Newton steps required, for a fixed problem instance. Check that your computed $x*$ and ν^* (nearly) satisfy the KKT conditions.

To generate some random problem data (i.e., A, b, c, x_0), we recommend the following approach. First, generate A randomly. (You might want to check that it has full rank.) Then generate a random positive vector x_0, and take $b = Ax_0$. (This ensures that $x0$ is strictly feasible.) The parameter c can be chosen randomly. To be sure the sublevel sets are bounded, you can add a row to A with all positive elements. If you want to be able to repeat a run with the same problem data, be sure to set the state for the uniform and normal random number generators.

Here are some hints that may be useful.

- We recommend computing λ^2 using the formula $\lambda^2 = -\Delta x_{nt} \nabla f(x)$.. You don't really need λ for anything; you can work with λ^2 instead. (This is important for reasons described below.)

- There can be small numerical errors in the Newton step $-\Delta x_{nt}$ that you compute. When x is nearly optimal, the computed value of λ^2, i.e., $\lambda^2 = -\Delta x_{nt} \nabla f(x)$, can actually be (slightly) negative. If you take the squareroot to get λ, you'll get a complex number, and you'll never recover. Moreover, your line search will never exit. However, this only happens when x is nearly optimal. So if you exit on the condition $\lambda^2/2 \leq 10^{-6}$, everything will be fine, even when the computed value of λ^2 is negative.

- For the line search, you must first multiply the step size t by $beta$ until $x+t\Delta x_{nt}$ is feasible (i.e., strictly positive). If you don't, when you evaluate f you'll be taking the logarithm of negative numbers, and you'll never recover.

2. LP solver with strictly feasible starting point. Using the centering code from part (1), implement a barrier method to solve the standard form LP

$$\text{minimize} \qquad c^T x - \sum_{i=1}^{n} log(x_i)$$
$$\text{subject to} \qquad Ax = b,$$

with variable $x \in R^n$, given a strictly feasible starting point x_0. Your LP solver should take as argument A, b, c, and x_0, and return x^*.

You can terminate your barrier method when the duality gap, as measured by n/t, is smaller then 10^{-3}. (If you make the tolerance much smaller, you might run into some numerical trouble.) Check your LP solver against the solution found by cvx, for several problem instances.

The comments in part (1) on how to generate random data hold here too.

Plot the progress of the algorithm, for a problem instance with $n = 500$ and $m = 100$, showing duality gap (on a log scale) on the vertical axis, versus the cumulative total number of Newton steps (on a linear scale) on the horizontal axis.

Your algorithm should return a $2 \times k$ matrix history, (where k is the total number of centering steps), whose first row contains the number of Newton steps required for each centering step, and whose second row shows the duality gap at the end of each centering step. In order to get a plot that looks like the ones in the book (e.g., figure 11.4, page 572), you should use the following code:

```
[xx, yy] = stairs(cumsum(history(1,:)),history(2,:));
semilogy(xx,yy);
```

3. LP solver. Using the code from part (2), implement a general standard form LP solver, that takes arguments A, b, c, determines (strict) feasibility, and returns an optimal point if the problem is (strictly) feasible.

You will need to implement a phase I method, that determines whether the problem is strictly feasible, and if so, finds a strictly feasible point, which can then be fed to the code from part (2). In fact, you can use the code from part (2) to implement the phase I method.

To find a strictly feasible initial point x0, we solve the phase I problem

$$\text{minimize} \qquad t$$
$$\text{subject to} \qquad Ax = b,$$
$$x \succeq (1 - t)\mathbf{1}, \quad t \geq 0,$$

with variables x and t. If we can find a feasible (x, t), with $t < 1$, then x is strictly feasible for the original problem. The converse is also true, so the original LP is strictly feasible if and only if $t^* < 1$, where t^* is the optimal value of the phase I problem.

We can initialize x and t for the phase I problem with any x_0 satisfying $Ax_0 = b$, and $t_0 = 2 - min_i x_i^0$, (Here we can assume that $min x_i^0 \le 0$; otherwise x_0 is already a strictly feasible point, and we are done.) You can use a change of variable $z = x + (t - 1)\mathbf{1}$ to transform the phase I problem into the form in part (2).

Check your LP solver against cvx on several numerical examples, including both feasible and infeasible instances.

Solution