Numerical Optimization

Lecture 6:   Sensitivity Analysis

王浩

信息科学与技术学院

Email: wanghao1@shanghaitech.edu.cn

本节内容

- 灵敏度分析
  - 增加单个变量
  - 增加单个约束（等式、不等式）
  - 改变右端向量某分量
  - 改变目标某系数
  - 改变系数矩阵某分量
  - 影子价格和既约费用

Command Window

```
>> load sc50b
>> options = optimoptions(@linprog,'Algorithm','dual-simplex','Display','iter');
>> ub = [];
>> [x,fval,exitflag,output,lambda] = linprog(f,A,b,Aeq,beq,lb,ub,options);

LP preprocessing removed 2 inequalities, 16 equalities,
16 variables, and 26 non-zero elements.

 Iter      Time            Fval  Primal Infeas    Dual Infeas
    0     0.001    0.000000e+00   0.000000e+00   1.305012e+00
    7     0.001   -1.587054e+02   3.760622e+02   0.000000e+00
   34     0.001   -7.000000e+01   0.000000e+00   0.000000e+00

Optimal solution found.

>> exitflag

exitflag =

     1

>> output

output =

  struct with fields:

        iterations: 34
    constrviolation: 1.7053e-13
           message: 'Optimal solution found.'
         algorithm: 'dual-simplex'
      firstorderopt: 1.2590e-14

>> lambda

lambda =

  struct with fields:

      lower: [48×1 double]
      upper: [48×1 double]
      eqlin: [20×1 double]
    ineqlin: [30×1 double]
```

```
>> lambda

lambda =

  struct with fields:

      lower: [48×1 double]
      upper: [48×1 double]
      eqlin: [20×1 double]
    ineqlin: [30×1 double]

>> lambda.eqlin

ans =

    0.1750
    0.1750
    0.1750
    0.1750
    0.7500
    0.1312
    0.1313
    0.1313
    0.1313
    0.5625
    0.0984
    0.0984
    0.0984
    0.0984
    0.4219
    0.0738
    0.0738
    0.0738
    0.0738
    0.3164
```

Sensitivity Analysis

- In many real-world problem, the following can occur:
  - The input data is not very accurate
  - We don’t know all of the constraints ahead of time
  - We don’t know all the variables ahead of time.
- Because of this, we want to analyze the dependence of the model on the input data, i.e.,
  - the matrix A
  - the right-hand side vector b, and
  - the cost vector c
- We would also like to know the effect of additional variables and constraints
- This is done using **sensitivity analysis**

The Fundamental Idea

- Using the simplex algorithm to solve a standard form problem, we know that if B is an optimal basis, then two conditions are satisfied:
  - $B^{-1}b \geq 0$
  - $c^T - c_B^T B^{-1} A \geq 0$
- When the problem is changed, we can check to see how these conditions are affected.
- This is the simplest kind of analysis.
- When using the simplex method, we always have B^{-1} available, so we can easily recompute appropriate quantities
- Where is B^{-1} in the simplex tableau?

Adding a New Variable

- Suppose we want to consider adding a new variable to the problem, e.g., we want to consider a new product to our line

- We simply compute the reduced cost of the new variable as

$$c_j - c_B^T B^{-1} A_j$$

  where A_j is the column corresponding to the new variable in the matrix.

- If the reduced cost is nonnegative, then we should not consider adding the product

- Otherwise, it is eligible to enter the basis and we can reoptimize from the current feasible (yet now non-optimal) basis (using which method?)

Adding a New Inequality Constraint

- Assume the new constraint is NOT satisfied by the current optimal solution (what if it is satisfied?)
- Suppose we want to introduce a new constraint of the form

$$a_{m+1}^T x \geq b_{m+1}$$

- The new constraint matrix (in standard form) would look like

$$\begin{bmatrix} A & 0 \\ a_{m+1}^T & -1 \end{bmatrix}$$

- Hence, the new basis matrix would look like

$$\bar{B} = \begin{bmatrix} B & 0 \\ a^T & -1 \end{bmatrix}$$

- The new basis inverse would then be

$$\bar{B}^{-1} = \begin{bmatrix} B^{-1} & 0 \\ a^T B^{-1} & -1 \end{bmatrix}$$

Adding a New Inequality Constraint

- The vector of reduced costs is

$$[c^T \ 0] - [c_B^T \ 0] \begin{bmatrix} B^{-1} & 0 \\ a^T B^{-1} & -1 \end{bmatrix} \begin{bmatrix} A & 0 \\ a_{m+1}^T & -1 \end{bmatrix} = [c^T - c_B^T B^{-1} A, \, 0]$$

  and so the reduced costs remain unchanged.

- Hence, we have a dual feasible basis and we apply dual simplex.

- The tableau can be computed as

$$\bar{B}^{-1} \begin{bmatrix} A & 0 \\ a_{m+1}^T & -1 \end{bmatrix} = \begin{bmatrix} B^{-1}A & 0 \\ a^T B^{-1} A - a_{m+1}^T & 1 \end{bmatrix}$$

- Note that $B^{-1}A$ is available from the original tableau

Adding a New Equality Constraint

- Assume the new constraint is NOT satisfied by the current optimal solution
- We introduce an artificial variable x_{n+1}, as in the two-phase method, and consider the LP (assuming $a_{m+1}^T x^* > b_{m+1}$)

$$
\begin{aligned}
\min \quad & c^T x + M x_{n+1} \\
\text{s.t.} \quad & Ax = b \\
& a_{m+1}^T x - x_{n+1} = b_{m+1} \\
& x \geq 0,\ x_{n+1} \geq 0
\end{aligned}
$$

- We can obtain a primal feasible basis by making the new variable basic.
- The new tableau can be computed as before.
- If the new problem is feasible and M is large enough, then the solution will have $x_{n+1} = 0$
- The values of the remaining variables will yield an optimal solution to the original problem with the additional constraint

Changes to the Right-hand Side

- Suppose we change b_i to $b_i + \delta$
- The values of the basic variables change from $B^{-1}b$ to $B^{-1}(b + \delta e^i)$, where e^i is the ith unit vector (we don't need to check reduced cost, why?)
- The feasibility condition is then

$$B^{-1}(b + \delta e^i) \geq 0$$

- If g is the ith column of B^{-1}, then the feasibility condition becomes

$$x_B + \delta g \geq 0$$

- This is equivalent to

$$\max_{\{j|g_j>0\}} \left(-\frac{x_{B(j)}}{g_j} \right) \leq \delta \leq \min_{\{j|g_j<0\}} \left(-\frac{x_{B(j)}}{g_j} \right)$$

- If δ is outside the allowable range, we can reoptimize using dual simplex

Changes in the Cost Vector

- Suppose we change some cost coefficient from c_j to $c_j + \delta$
- If c_j is the cost coefficient of a nonbasic variable, then we need recalculate its reduced cost $c_i - c_B B^{-1} a_j + \delta$
- The reduced cost itself increases by δ and the current solution remains optimal as long as $\delta \geq -r_j$
- If c_l is the cost coefficient of the lth basic variable, then c_B becomes $c_B + \delta e_l$ and the new optimality conditions are

$$(c_B + \delta e_l)^T B^{-1} A \leq c^T$$

- This is equivalent to

$$\delta q \leq r$$

where q is the lth row of $B^{-1}A$, which is available in the simplex tableau

Changes in a Nonbasic Column of A

- Suppose we change some entry a_{ij} from the constraint matrix to $a_{ij} + \delta$
- If column j is nonbasic, then B does not change and we only need to check the reduced cost of column j
- The new reduced cost is

$$c_j - c_B^T B^{-1}(A_j + \delta e^i)$$

- This means the current solution remains optimal if

$$r_j - \delta p_i \geq 0$$

- Otherwise, we reoptimize with primal simplex

Changes in a Basic Column of A

- This case is more complicated; the tragedy case
- $B + \delta e^i$ may not be a basis anymore;
- $B + \delta e^i$ may even not invertible
- $B + \delta e^i$ is invertible, but $(B + \delta e^i)^{-1}$way different from B^{-1}
- Generally we have to reoptimize, may starting from the beginning

Sensitivity and Shadow Price

$$\begin{array}{rll} z(b) = & \textbf{minimize} & c^{\mathrm{T}} x \\ & \textbf{subject to} & A x = b \\ & & x \geq 0 \end{array}$$

- Suppose the optimal basis matrix is B with $B^{-1}b > 0$ (nondengenerate)

$$\lambda^T = c_B^T B^{-1}$$

- Question: if b changes a little bit, how would the optimal value change?

$$b \leftarrow b + \delta \implies z(\delta) - z = ?$$

- Let $\frac{\partial}{\partial b_j} z(b)$ denote the partial derivative of $z(b)$ w.r.t. b_j

$$\frac{\partial}{\partial b_j} z(b) = \lambda_j$$

- We call λ_j is the marginal price, or shadow price, associated with b_j

Sensitivity and Reduced Cost

- Suppose we change some cost coefficient from c_j to $c_j + \delta$
- If c_j is the cost coefficient of a nonbasic variable, then we need recalculate its reduced cost $c_i - c_B B^{-1} a_j + \delta$
- The reduced cost itself increases by δ and the current solution remains optimal as long as $\delta \geq - r_j$
- If you reduce c_j more than r_j, the current basis must change, and x_j may enter the basis