L9 supplementary lecture: initial simplex for Nelder Mead

Yanfei Kang

3/23/2020

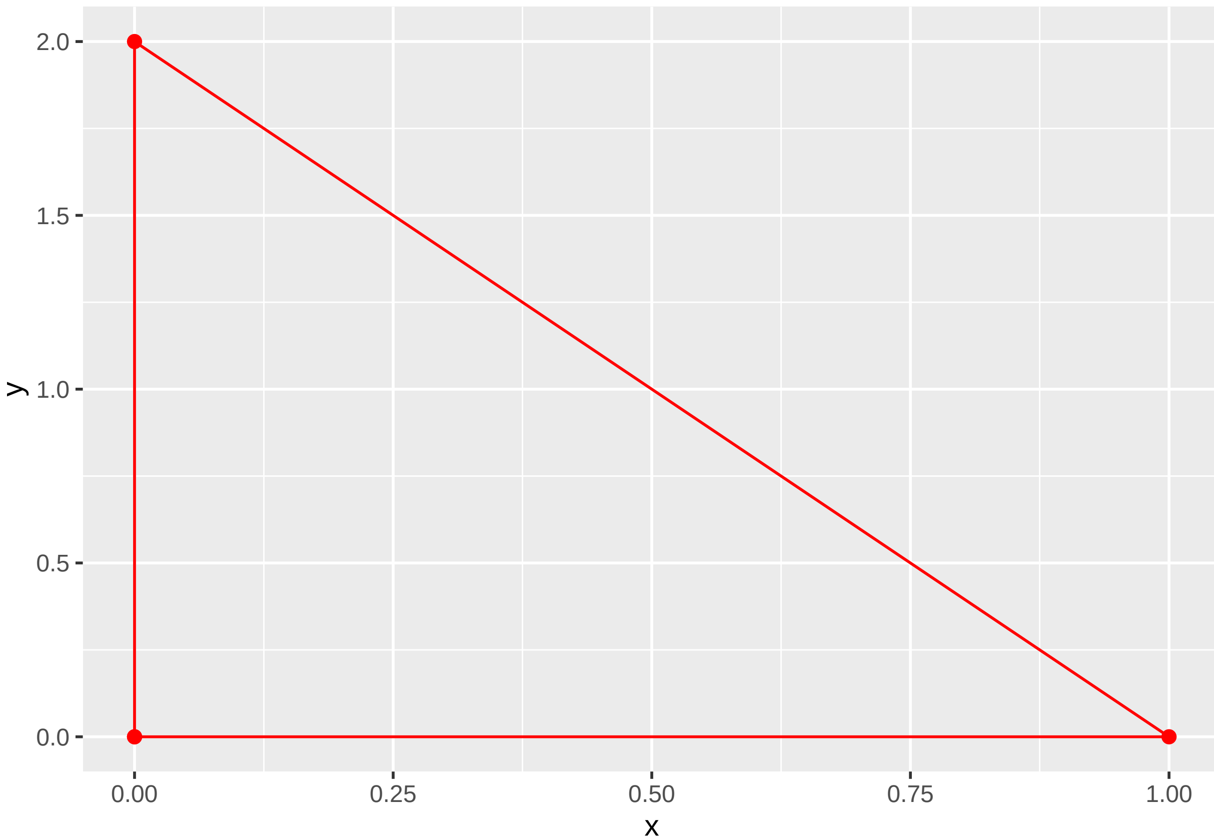
Nelder Mead algorithm is sensitive to initial simplex. Determining the initial simplex is hard for large n. There are some approaches to compute the initial simplex. But no initial simplex actually clearly improves on the others, in practice, it may be convenient to try different approaches.

Axis-by-axis simplex

- We now define the vertices of the simplex $S = \{\mathbf{x_i}\}_{i=1,\cdots,n+1}$.
- This simplex depends on a vector of positive lengths l.
- The first vertex of the simplex is the initial guess $\mathbf{x_0}$.
- The other vertices are defined by

$$(\mathbf{v}_i)_j = \begin{cases} (\mathbf{x}_0)_j + \ell_j, \text{ if } j = i-1 \\ (\mathbf{x}_0)_j, \text{ if } j \neq i-1 \end{cases}$$

  for vertices $i = 2, \cdots, n+1$ and components $j = 1, \cdots, n$.



Spendley's et al regular simplex

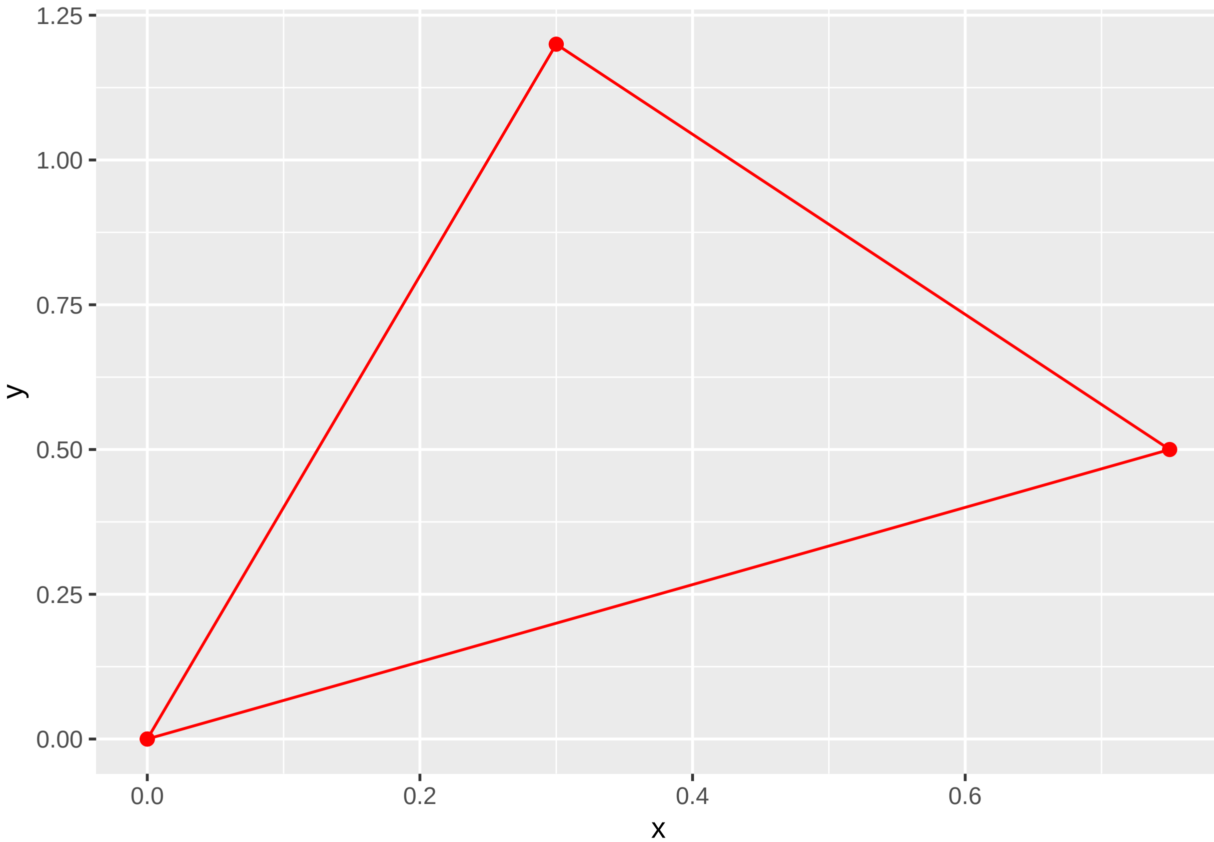
- Spendley, Hext and Himsworth use a regular simplex with given size $l > 0$. We define the parameters $p, q > 0$ as

$$p = \frac{1}{n\sqrt{2}}(n - 1 + \sqrt{n+1})$$
$$q = \frac{1}{n\sqrt{2}}(\sqrt{n+1} - 1)$$

- The first vertex of the simplex is the initial guess $\mathbf{x_0}$.
- The other vertices are defined by

$$(\mathbf{v}_i)_j = \begin{cases} (\mathbf{x}_0)_j + \ell p, & \text{if } j = i - 1 \\ (\mathbf{x}_0)_j + \ell q, & \text{if } j \neq i - 1 \end{cases}$$

  for vertices $i = 2, \cdots, n+1$ and components $j = 1, \cdots, n$.



Random bounds

- Assume that the variable $\mathbf{x}$ is bounded by $m_j \leq x_j \leq M_j$.
- The other vertices can be obtained by

$$(\mathbf{v}_i)_j = m_j + \theta_{i,j} \left(M_j - m_j \right),$$

  where $\theta_{i,j} \in [0, 1]$.

Pfeffer's method

R packages

- You can find several R packages dealing with this problem.
- `optimsimplex`, `neldermead`, etc.
- You may try to improve your own functions.