A new algorithm for continuous optimization

Tejas A Desai*

The Indian Institute of Management, Vastrapur, Ahmedabad--380015, Gujarat, India

Abstract

We present a new algorithm for continuous, nonlinear or linear, and constrained or unconstrained optimization. After proving its convergence, we apply it to unconstrained and constrained maximum likelihood estimation, and compare its performance to that of the Newton-Raphson algorithm.

Keywords: Nonlinear programming; Linear programming; Constraints satisfaction; Multivariate statistics; Simulation

1. Introduction

A vast amount of literature exists on both non-linear as well as linear continuous optimization. Many key references in this area are available, for example, in Luenberger (2004). However, to the best of our knowledge, the approach discussed in this paper has not been considered before.

To motivate the algorithm, in Section 2.1 we first consider the problem of finding a solution to a set of nonlinear and/or linear inequalities. After describing an algorithm for

*Tel.: +91 79 2632 4937; fax.: +91 79 2630 6896

Email address: tdesai@iimahd.ernet.in

this problem, we prove its convergence to a solution, and apply it to an introductory example. Thereafter, in Section 2.2, we show how the problem of continuous and possibly constrained optimization, namely, the problem of maximizing a continuous objective function in possible presence of nonlinear or linear constraints, may be considered an extension of the first problem. An extended algorithm is proposed for this problem, and its convergence is proved. The extended algorithm is then applied to unconstrained maximum likelihood estimation in presence of imcomplete data. For the sake of comparision, the Newton-Raphson method is also applied to this example. Next, the (extended) algorithm is applied to an example of constrained maximum likelihood estimation, and its performance is also compared to that of the Newton-Raphson method.

2. The algorithm

2.1 Motivation: Finding a solution to a system of linear and/or non-linear inequalities

Consider the following system of inequalities:

$$h_j(x_1, \ldots, x_q) \leq a_j,\ j = 1, \ldots, p;\ \text{and} \tag{1}$$

$$b_i \leq x_i \leq c_i,\ i = 1, \ldots, q$$

Let

$$\boldsymbol{B} = [b_1, c_1] \times \ldots \times [b_q, c_q]$$

Any of the first p inequalities above may be linear or nonlinear. Let $\boldsymbol{x} = (x_1, \ldots, x_q)$ denote a point in $\mathbb{R}^{\boldsymbol{q}}$. Let

$$\boldsymbol{A_j} = \left\{ \boldsymbol{x} \in \mathbb{R}^q \middle| \boldsymbol{x} \in \boldsymbol{B}, h_j(\boldsymbol{x}) \leq a_j \right\},\ j = 1, \ldots, p;$$

and

$$\boldsymbol{S} = \left(\bigcap_{j=1}^{p} \boldsymbol{A_j} \right)$$

Then finding a solution to system (1) above is equivalent to finding a point in the set $\boldsymbol{S}$. Our algorithm is denoted **A1**. To start **A1**, draw $\boldsymbol{x^{(o)}}$ randomly from $\boldsymbol{B}$. If $\boldsymbol{x^{(o)}}$ satisfies the p inequalities in (1), then stop the algorithm. Else, in the next iteration draw $\boldsymbol{x^{(1)}}$ randomly from $\boldsymbol{B}$. If $\boldsymbol{x^{(1)}}$ satisfies the p inequalities in (1), then stop the algorithm; else, go to iteration 2. In general, suppose **A1** has not converged at iteration $i \geq 0$. Then in the iteration $i+1$, draw randomly $\boldsymbol{x^{(i+1)}}$ from $\boldsymbol{B}$. If $\boldsymbol{x^{(i+1)}}$ satisfies the p inequalities in (1), then stop the algorithm; else, go to iteration $i+2$. Then we have the following result:

Theorem 2.1. *Suppose that the set S defined above has positive Lebesgue measure in $\mathbb{R}^q$. Then the algorithm* **A1** *described above converges to a solution in S in a finite number of iterations with probability 1.*

Proof. The result is a direct consequence of the fact that $\boldsymbol{S}$ is a set of positive Lebesgue measure in $\mathbb{R}^{\boldsymbol{q}}$**.** In other words, the probability that a point randomly sampled from $\boldsymbol{B}$ is in $\boldsymbol{S}$ is positive. Thus, the event that a sequence $\{\boldsymbol{x^{(i)}}\}$ generated by the algorithm **A1** fails to converge in a finite number of iterations is of probability 0. □

Now we apply **A1** to the following example.

Example 2.1. Consider the following system of inequalities:

$$sin(x) - y \leq 0,$$

$$y - e^x \leq 0,$$

$$x^3 - y \leq 0, \text{ and}$$

$$-10 \leq x, y \leq 10$$

Algorithm **A1** was run 100 times to find a solution to the above system of inequalities. The algorithm converged to a solution each of the 100 times. A summary of the real times required by **A1** to converge is presented in Table 1 below.

$Table\,1$: A summary of the number of iterations, the real times required by **A1**	
	Real time (in seconds)
Max	0.51
Mean	0.02
Median	0.01
Minimum	0.01

In the next section, we extend algorithm **A1** to the case of continuous nonlinear or linear, constrained or unconstrained optimization.

2.2 Continuous nonlinear or linear, constrained or unconstrained optimization

Consider the following optimization problem :

$$\text{Maximize } f(x_1, \ldots, x_q) \text{ subject to} \tag{2}$$

$$h_j(x_1, \ldots, x_q) \leq a_j,\ j = 1, \ldots, p, \text{ and}$$

$$b_i \leq x_i \leq c_i,\ i = 1, \ldots, q.$$

The last $p + q$ constraints may or may not be there. Furthermore, *if we assume that the function f is continuously differentiable over the domain implicit in* (2), *achieves exactly one extremum in the interior of the domain and that this extremum is a maximum*, then finding a solution to the problem (2) reduces to finding solutions to the following system of inequalities:

$$\frac{\partial f}{\partial x_1} \leq 0, \ldots, \frac{\partial f}{\partial x_q} \leq 0 \tag{3}$$

$$-\frac{\partial f}{\partial x_1} \leq 0, \ldots, -\frac{\partial f}{\partial x_q} \leq 0$$

$$h_j(x_1, \ldots, x_q) \leq a_j,\ j = 1, \ldots, p; \text{ and}$$

$$b_i \leq x_i \leq c_i,\ i = 1, \ldots, q$$

Let $\boldsymbol{B}$ be defined as in the previous section. For each component j, $1 \leq j \leq q$, consider the two relations of $\frac{\partial f}{\partial x_j}$ with 0: $\frac{\partial f}{\partial x_j} \leq 0$ and $\frac{\partial f}{\partial x_j} \geq 0$. Consider the following $2 \cdot q$ regions $\boldsymbol{A_{m,k}}$ in $\mathbb{R}^{\boldsymbol{q}}$, $1 \leq m \leq q$, $1 \leq k \leq 2$:

$$\boldsymbol{A_{m,1}} = \left\{ \boldsymbol{x} \in \mathbb{R}^q \middle| \frac{\partial f}{\partial x_m} \leq 0, \, \boldsymbol{x} \in \boldsymbol{B}, h_j(\boldsymbol{x}) \, \leq a_j, j = 1, \ldots, p \right\}, \text{ and}$$

$$\boldsymbol{A_{m,2}} = \left\{ \boldsymbol{x} \in \mathbb{R}^q \middle| \frac{\partial f}{\partial x_m} \geq 0, \, \boldsymbol{x} \in \boldsymbol{B}, h_j(\boldsymbol{x}) \, \leq a_j, j = 1, \ldots, p \right\}, \, m = 1, \ldots, q$$

We will assume that each of the 2^q sets

$$\boldsymbol{A_{1,k_1}} \bigcap \boldsymbol{A_{2,k_2}} \ldots \bigcap \boldsymbol{A_{q-1,k_{q-1}}} \bigcap \boldsymbol{A_{q,k_q}}, \, 1 \leq k_1, \ldots, k_q \leq 2 \tag{4}$$

has positive Lebesgue measure in $\mathbb{R}^{\boldsymbol{q}}$. The algorithm for finding the solution to system (3) will be denoted **A2**. The first iteration of **A2**, denoted by $i = 1$, is as follows. We randomly select 2^q points

$$\boldsymbol{x}_{\boldsymbol{k_1^*,j}}^{\boldsymbol{(i,1)}} \in \boldsymbol{D}_{\boldsymbol{k_1^*,j}}^{\boldsymbol{(1)}} = \left(\boldsymbol{A_{1,k_1}} \bigcap \boldsymbol{A_{2,k_2}} \ldots \bigcap \boldsymbol{A_{q-1,k_{q-1}}} \right) \bigcap \boldsymbol{A_{q,j}}, \, 1 \leq j \leq 2, \, \boldsymbol{k_1^*} = (\boldsymbol{k_1}, \ldots, \boldsymbol{k_{q-1}}).$$

where $1 \leq k_1, \ldots, k_{q-1} \leq 2$, $1 \leq j \leq 2$. This selection is done by applying algorithm **A1** to each of the 2^q sets $\boldsymbol{D}_{\boldsymbol{k_1^*,j}}^{\boldsymbol{(1)}}$. Consider the average $(\boldsymbol{x}_{\boldsymbol{k_1^*,1}}^{\boldsymbol{(i,1)}} + \boldsymbol{x}_{\boldsymbol{k_1^*,2}}^{\boldsymbol{(i,1)}})/2$. If this average lies in $\boldsymbol{D}_{\boldsymbol{k_1^*,1}}^{\boldsymbol{(1)}}$then let $\boldsymbol{x}_{\boldsymbol{k_1^*,1}}^{\boldsymbol{(i,1)}} = (\boldsymbol{x}_{\boldsymbol{k_1^*,1}}^{\boldsymbol{(i,1)}} + \boldsymbol{x}_{\boldsymbol{k_1^*,2}}^{\boldsymbol{(i,1)}})/2$. Similarly, if this average lies in $\boldsymbol{D}_{\boldsymbol{k_1^*,2}}^{\boldsymbol{(1)}}$ then let $\boldsymbol{x}_{\boldsymbol{k_1^*,2}}^{\boldsymbol{(i,1)}} = (\boldsymbol{x}_{\boldsymbol{k_1^*,1}}^{\boldsymbol{(i,1)}} + \boldsymbol{x}_{\boldsymbol{k_1^*,2}}^{\boldsymbol{(i,1)}})/2$.. Repeat this procedure for all the 2^{q-1} possible values of $\boldsymbol{k_1^*}$ until $\boldsymbol{x}_{\boldsymbol{k_1^*,1}}^{\boldsymbol{(i,1)}} = \boldsymbol{x}_{\boldsymbol{k_1^*,2}}^{\boldsymbol{(i,1)}}$. Now let $\boldsymbol{x}_{\boldsymbol{k_2^*,1}}^{\boldsymbol{(i,2)}} = \boldsymbol{x}_{\boldsymbol{k_1^*,1}}^{\boldsymbol{(i,1)}}$ if $k_{q-1} = 1$ and $\boldsymbol{x}_{\boldsymbol{k_2^*,2}}^{\boldsymbol{(i,2)}} = \boldsymbol{x}_{\boldsymbol{k_1^*,1}}^{\boldsymbol{(i,1)}}$ if $k_{q-1} = 2$. Note that $\boldsymbol{x}_{\boldsymbol{k_2^*,1}}^{\boldsymbol{(i,2)}} \in \boldsymbol{D}_{\boldsymbol{k_2^*,1}}^{\boldsymbol{(2)}}$ and $\boldsymbol{x}_{\boldsymbol{k_2^*,2}}^{\boldsymbol{(i,2)}} \in \boldsymbol{D}_{\boldsymbol{k_2^*,2}}^{\boldsymbol{(2)}}$ where

$$\boldsymbol{D}_{\boldsymbol{k_2^*,j}}^{\boldsymbol{(2)}} = \left(\boldsymbol{A_{1,k_1}} \bigcap \boldsymbol{A_{2,k_2}} \ldots \bigcap \boldsymbol{A_{q-2,k_{q-2}}} \right) \bigcap \boldsymbol{A_{q-1,j}}, \, 1 \leq j \leq 2, \, \boldsymbol{k_2^*} = (\boldsymbol{k_1}, \ldots, \boldsymbol{k_{q-2}}),$$

where $1 \leq k_1, \ldots, k_{q-2} \leq 2$, $1 \leq j \leq 2$. Consider the average $(\boldsymbol{x}_{\boldsymbol{k_2^*,1}}^{\boldsymbol{(i,2)}} + \boldsymbol{x}_{\boldsymbol{k_2^*,2}}^{\boldsymbol{(i,2)}})/2$. If this average lies in $\boldsymbol{D}_{\boldsymbol{k_2^*,1}}^{\boldsymbol{(2)}}$then let $\boldsymbol{x}_{\boldsymbol{k_2^*,1}}^{\boldsymbol{(i,2)}} = (\boldsymbol{x}_{\boldsymbol{k_2^*,1}}^{\boldsymbol{(i,2)}} + \boldsymbol{x}_{\boldsymbol{k_2^*,2}}^{\boldsymbol{(i,2)}})/2$. Similarly, if this average lies in $\boldsymbol{D}_{\boldsymbol{k_2^*,2}}^{\boldsymbol{(2)}}$ then let $\boldsymbol{x}_{\boldsymbol{k_2^*,2}}^{\boldsymbol{(i,2)}} = (\boldsymbol{x}_{\boldsymbol{k_2^*,1}}^{\boldsymbol{(i,2)}} + \boldsymbol{x}_{\boldsymbol{k_2^*,2}}^{\boldsymbol{(i,2)}})/2$.. Repeat this procedure for all the 2^{q-2}

possible values of $\boldsymbol{k_2^*}$, until $\boldsymbol{x_{k_2^*,1}^{(i,2)}} = \boldsymbol{x_{k_2^*,2}^{(i,2)}}$. Now let $\boldsymbol{x_{k_3^*,1}^{(i,3)}} = \boldsymbol{x_{k_2^*,1}^{(i,2)}}$ if $k_{q-2} = 1$ and $\boldsymbol{x_{k_3^*,2}^{(i,3)}} = \boldsymbol{x_{k_2^*,1}^{(i,2)}}$ if $k_{q-2} = 2$. Note that $\boldsymbol{x_{k_3^*,1}^{(i,3)}} \in \boldsymbol{D_{k_3^*,1}^{(3)}}$ and $\boldsymbol{x_{k_3^*,2}^{(i,3)}} \in \boldsymbol{D_{k_3^*,2}^{(3)}}$ where

$$\boldsymbol{D_{k_3^*,j}^{(3)}} = \Big(\boldsymbol{A_{1,k_1}} \bigcap \boldsymbol{A_{2,k_2}} \ldots \bigcap \boldsymbol{A_{q-3,k_{q-3}}}\Big) \bigcap \boldsymbol{A_{q-2,j}},\ 1 \leq j \leq 2,\ \boldsymbol{k_3^*} = (\boldsymbol{k_1}, \ldots, \boldsymbol{k_{q-3}}),$$

We repeat the above averaging processes until we arrive at the points $\boldsymbol{x_{k_1,1}^{(i,q-1)}}$ and $\boldsymbol{x_{k_1,2}^{(i,q-1)}}$ such that $\boldsymbol{x_{k_1,1}^{(i,q-1)}} \in \boldsymbol{D_{k_{q-1}^*,1}^{(q-1)}} = \boldsymbol{A_{1,k_1}} \bigcap \boldsymbol{A_{2,1}}$ and $\boldsymbol{x_{k_1,2}^{(i,q-1)}} \in \boldsymbol{D_{k_{q-1}^*,2}^{(q-1)}} = \boldsymbol{A_{1,k_1}} \bigcap \boldsymbol{A_{2,2}}$. Here $\boldsymbol{k_{q-1}^*} = k_1$. Consider the average $(\boldsymbol{x_{k_1,1}^{(i,q-1)}} + \boldsymbol{x_{k_2,2}^{(i,q-1)}})/2$. If this average lies in $\boldsymbol{D_{k_{q-1}^*,1}^{(q-1)}}$ then let $\boldsymbol{x_{k_1,1}^{(i,q-1)}} = (\boldsymbol{x_{k_1,1}^{(i,q-1)}} + \boldsymbol{x_{k_1,2}^{(i,q-1)}})/2$. Similarly, if this average lies in $\boldsymbol{D_{k_{q-1}^*,2}^{(q-1)}}$ then let $\boldsymbol{x_{k_1,2}^{(i,q-1)}} = (\boldsymbol{x_{k_1,1}^{(i,q-1)}} + \boldsymbol{x_{k_1,2}^{(i,q-1)}})/2$. Repeat this procedure for the 2 possible values of $\boldsymbol{k_1}$, until $\boldsymbol{x_{k_1,1}^{(i,q-1)}} = \boldsymbol{x_{k_1,2}^{(i,q-1)}}$. Now let $\boldsymbol{x_1^{(i,q)}} = \boldsymbol{x_{1,1}^{(i,q-1)}}$ if $k_1 = 1$ and $\boldsymbol{x_2^{(i,q)}} = \boldsymbol{x_{2,1}^{(i,q-1)}}$ if $k_1 = 2$. Consider the average $(\boldsymbol{x_1^{(i,q)}} + \boldsymbol{x_2^{(i,q)}})/2$. If this average lies in $\boldsymbol{A_{1,1}}$ then let $\boldsymbol{x_1^{(i,q)}} = (\boldsymbol{x_1^{(i,q)}} + \boldsymbol{x_2^{(i,q)}})/2$. If the average lies in $\boldsymbol{A_{1,2}}$ then let $\boldsymbol{x_2^{(i,q)}} = (\boldsymbol{x_1^{(i,q)}} + \boldsymbol{x_2^{(i,q)}})/2$. We will denote the average $(\boldsymbol{x_1^{(i,q)}} + \boldsymbol{x_2^{(i,q)}})/2$ as $\boldsymbol{w^{(i)}}$. This ends iteration i.

Iteration $i+1$ is similar to iteration i, except that using **A1** we randomly select 2^q points

$$\boldsymbol{x_{k_1^*,j}^{(i+1,1)}} \in \boldsymbol{D_{k_1^*,j}^{(1)}} = \Big(\boldsymbol{A_{1,k_1}} \bigcap \boldsymbol{A_{2,k_2}} \ldots \bigcap \boldsymbol{A_{q-1,k_{q-1}}}\Big) \bigcap \boldsymbol{A_{q,j}},\ 1 \leq j \leq 2,\ \boldsymbol{k_1^*} = (\boldsymbol{k_1}, \ldots, \boldsymbol{k_{q-1}})$$

such that the first component of $\boldsymbol{x_{k_1^*,j}^{(i+1,1)}}$ *, $1 \leq j \leq 2$, is the same as the first component of* $\boldsymbol{x_1^{(i,q)}}$ *if* $k_1 = 1$ *and same as the first component of* $\boldsymbol{x_2^{(i,q)}}$ *if* $k_1 = 2$. The rest of the iteration $i+1$ is the same as iteration i, except that $\boldsymbol{x_1^{(i,q)}}$ and/or $\boldsymbol{x_2^{(i,q)}}$ will get updated at the end of iteration $i+1$ to $\boldsymbol{x_1^{(i+1,q)}}$ and/or $\boldsymbol{x_2^{(i+1,q)}}$. Then in iteration $i+2$, the first component of $\boldsymbol{x_{k_1^*,j}^{(i+2,1)}}$, $1 \leq$ j ≤ 2, is the same as the first component of $\boldsymbol{x_1^{(i+1,q)}}$ if $k_1 = 1$ and same as the first component of $\boldsymbol{x_2^{(i+1,q)}}$ if $k_1 = 2$; and so on for subsequent iterations. To ensure that each iteration finishes in finite time, we use an $\epsilon \in$ (0,0·1) such that if

$$|| \boldsymbol{x_{k_j^*,1}^{(i,j)}} - \boldsymbol{x_{k_j^*,2}^{(i,j)}} || < \epsilon,\ 1 \leq \text{j} \leq q,$$

then we consider $\boldsymbol{x}_{k_j^*,1}^{(i,j)}$ and $\boldsymbol{x}_{k_j^*,2}^{(i,j)}$ to be equal *upto the decimal point implied by* ϵ.

Then we have the following result:

Theorem 2.2. *Suppose the function* f *is continuously differentiable over the domain implied by the constraints in* (3)*, and has exactly one maximum and no other extremum in the domain implied by the constraints in* (3)*. Furthermore, suppose that this maximum occurs in the interior of the domain. Suppose each of the sets defined in* (4) *has positive Lebesgue measure in* $\mathbb{R}^q$*. Furthermore, suppose that each of the sets* $D_{k_a^*,l}^{(a)} \bigcup D_{k_a^*,2}^{(a)}$, $1 \leq a \leq q-1$*, and* $A_{1,1} \bigcup A_{1,2}$ *is convex in* $\mathbb{R}^q$*. Then the sequence of points* $\left\{w^{(i)}\right\}$ *converges to the solution* w *to the system (3) above as* $i \to \infty$.

Proof. The condition of positive Lebesgue measure ensures that algorithm **A1** converges in finite time at each iteration i. The convexity property ensures that each itreation and subiteration of **A2** is well-defined. Note that

$$||\boldsymbol{w}^{(i+1)} - \boldsymbol{w}^{(i)}|| < ||\boldsymbol{w}^{(i)} - \boldsymbol{w}^{(i-1)}||,\ i \geq 2$$

Thus the sequence $\left\{\boldsymbol{w}^{(i)}\right\}$ converges to a point $\boldsymbol{w}$ in $\mathbb{R}^{\boldsymbol{q}}$**.** Note that by construction,

$$\boldsymbol{w} \in \bigcap_{1 \leq k_1, \ldots, k_q \leq 2} (\boldsymbol{A_{1,k_1}} \bigcap \boldsymbol{A_{2,k_2}} \ldots \bigcap \boldsymbol{A_{q-1,k_{q-1}}} \bigcap \boldsymbol{A_{q,k_q}}).$$

Thus, $\boldsymbol{w}$ is the solution to (3); i.e., $\left.\frac{\partial f}{\partial x_k}\right|_{\boldsymbol{x}=\boldsymbol{w}} = 0,\ 1 \leq k \leq q$ (upto the decimal point implied by ϵ). □

Example 2.2. $Unconstrained\ Maximum\ Likelihood\ Estimation$

Consider a binomial random variable X such that $Pr(X=0) = 0.5$ and $Pr(X=1) = 0.5$. Consider a random variable Y such that if $X=0$, Y is distributed as $N(\mu_1 = 0, \sigma^2 = 1)$ and if $X=1$, Y is distributed as $N(\mu_1 = 1, \sigma^2 = 1)$. Furthermore, X is observed with probability 0.7, and X is randomly missing with probability 0.3. The variance σ^2 and $Pr(X=1)$ are assumed known, whereas the means μ_1 and μ_2 are

assumed unknown and are to be estimated. Table 2 presents 30 simulated values of X and Y:

$Table\ 2$: Data for Example 2.2		
i	x_i	y_i
1	.	0.35091
2	.	1.02329
3	.	-0.66084
4	0	0.73133
5	0	-1.40108
6	0	1.74854
7	0	1.45779
8	0	-0.32389
9	0	0.43005
10	0	-1.71751
11	0	-0.36244
12	0	-1.37992
13	0	0.25498
14	0	-0.35546
15	0	-0.69894
16	0	0.13100
17	0	1.42635
18	1	0.28686
19	1	1.62940
20	1	-0.46279
21	1	1.68601
22	1	0.68534
23	1	-0.72390
24	1	0.72166
25	1	2.78640
26	1	3.61448
27	1	1.18818
28	1	-0.55797
29	1	1.71294
30	1	-1.23507

We will denote the mle of $E(Y|X=0)$ as $\widehat{\mu}_1$ and the mle of $E(Y|X=1)$ as $\widehat{\mu}_2$. Let $f(y|\mu_i)$ denote the density of $N(\mu=\mu_i, \sigma^2=1)$, $i=1,2$. Let π denote the probability of X being missing. Furthermore, let $\boldsymbol{Y}$ denote the data in the third column of Table 2, and $\boldsymbol{X_{obs}}$ denote the observed data in the second column of Table 2. Last, let $I(X_i=0)$

and $I(X_i = 1)$, $i = 4$ to 30 be indicator functions indicating which values X_i takes, $i = 4$ to 30. Then the loglikelihood of μ_1,μ_2, and π given the observed data in Table 2 and known variance is as follows:

$$logL(\mu_1, \mu_2|\sigma^2, \boldsymbol{Y}, \boldsymbol{X_{obs}}) = \sum_{i=1}^{3} (log(0.5 \cdot f(y_i|\mu_1) + 0.5logf(y_i|\mu_2) + log\pi)$$

$$+ \sum_{i=4}^{30} (I(X_i = 0) \cdot logf(y_i|\mu_1) + I(X_i = 1) \cdot logf(y_i|\mu_2) + log(1 - \pi))$$

Then finding the maximum likelihood estimates of μ_1 and μ_2 is equivalent to the following non-linear, continuous, and unconstrained optimization problem:

$$\text{Maximize } logL(\mu_1, \mu_2|\sigma^2, \boldsymbol{Y}, \boldsymbol{X_{obs}}) \text{ .}$$

Algorithms **A2** and Newton-Raphson were each run 100 times to maximize the likelihood yielded by the observed data. Table 3 presents a summary of the values to which algorithm **A2** converges, along with a summary of the values to which the Newton-Raphson converges. Also presented are summaries of real times in seconds taken by each algorithm.

Table 3 : A summary of the values yielded and real times taken by **A2** and Newton-Raphson

Algorithm	**A2**			Newton-Raphson		
	$\widehat{\mu}_1$	$\widehat{\mu}_2$	real time (in secs.)	$\widehat{\mu}_1$	$\widehat{\mu}_2$	real time (in secs.)
Max	0.00	0.83	0.38	0.00	102.22	0.03
Mean	0.00	0.83	0.26	-0.26	1.84	0.01
Median	0.00	0.83	0.27	0.00	0.83	0.00
Minimum	0.00	0.83	0.22	-26.59	0.83	0.00

Table 3 demonstrates that while the Newton-Raphson algorithm is faster, it did not converge to the correct value (i.e., $\widehat{\mu}_1 = 0.00$ and $\widehat{\mu}_2 = 0.83$) 100 out of 100 times. Furthermore, although **A2** is slower compared to Newton-Raphson, the former is still quite fast in the sense that it took less than a second to converge each time.

Example 2.3. *Constrained maximum likelihood estimation*

Consider a bivariate normal random variable (X, Y) such that (X, Y) is distributed as $N(\boldsymbol{\mu_1}, \boldsymbol{\Sigma})$ with probability 0.5 (assumed known) and as $N(\boldsymbol{\mu_2}, \boldsymbol{\Sigma})$ with probability 0.5. The population indicator, i.e., the random variable which indicates whether an observation is from $N(\boldsymbol{\mu_1}, \boldsymbol{\Sigma})$ or from $N(\boldsymbol{\mu_2}, \boldsymbol{\Sigma})$ is not observed at all. The parameters are as follows:

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & -0.3 \\ -0.3 & 1 \end{pmatrix}, \boldsymbol{\mu_1} = (\mu_1, \mu_1) = (2, 2), \text{ and } \boldsymbol{\mu_2} = (\mu_2, \mu_2) = (-2, -2).$$

The covariance matrix $\boldsymbol{\Sigma}$ is assumed known, whereas μ_1 and μ_2 are assumed unknown and are to be estimated. Using the parameters above, 30 data were simulated. These data are presented in Table 4 :

$Table\ 4:$ Data for example 2.3		
i	x_i	y_i
1	-2.38477	-2.42155
2	-1.74090	-2.05489
3	3.08294	2.63114
4	3.36635	1.96981
5	2.54749	3.68319
6	-1.48511	-2.09576
7	-2.68046	-5.01516
8	-1.26134	-2.95281
9	1.05888	3.33043
10	2.17464	2.09240
11	4.17771	1.14555
12	-2.36698	-3.32461
13	-2.28839	-3.98008
14	2.07698	2.27375
15	2.13451	0.10032
16	-2.22344	-2.49873
17	1.94836	1.05031
18	-2.46237	-4.73239
19	1.76903	1.31959
20	-3.41411	-2.16737
21	-1.94223	-4.67921
22	0.47866	2.36898
23	-2.12063	-1.82635
24	-3.13627	-2.84237
25	-0.12150	-2.66558
26	1.23187	2.31169
27	2.17193	2.56457
28	-3.99247	-3.23134
29	2.82728	1.63710
30	2.25181	2.08827

Let $f(x, y|\boldsymbol{\mu_1}, \boldsymbol{\Sigma})$ and $f(x, y|\boldsymbol{\mu_2}, \boldsymbol{\Sigma})$ denote the densities of the $N(\boldsymbol{\mu_1}, \boldsymbol{\Sigma})$ and $N(\boldsymbol{\mu_2}, \boldsymbol{\Sigma})$ distributions, respectively. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ denote the data in the second and third columns of Table 4. Then the loglikelihood of $\boldsymbol{\mu_1}$ and $\boldsymbol{\mu_2}$ given $\boldsymbol{\Sigma}$, $\boldsymbol{X}$ and $\boldsymbol{Y}$ is

$$logL(\boldsymbol{\mu_1}, \boldsymbol{\mu_2}|\boldsymbol{\Sigma}, \boldsymbol{X}, \boldsymbol{Y}) = \sum_{i=1}^{30} log(0.5 \cdot f(x_i, y_i|\boldsymbol{\mu_1}, \boldsymbol{\Sigma}) + 0.5 f(x_i, y_i|\boldsymbol{\mu_2}, \boldsymbol{\Sigma}))$$

Then finding the maximum likelihood estimates of μ_1 and μ_2 subject to the constraints

$\mu_1 > \mu_2$ and $\mu_1 > 0$ is equivalent to solving the following nonlinear, continuous and constrained optimization problem:

$$\text{Maximize } logL(\mu_1, \mu_2|\boldsymbol{\Sigma}, \boldsymbol{X}, \boldsymbol{Y}) \text{ subject to}$$

$$\mu_1 > \mu_2 \text{ and}$$

$$\mu_1 > 0$$

Algorithms **A2** and Newton-Raphson were each run 100 times to maximize the likelihood yielded by the data. The Newton-Raphson algorithm not only failed to converge 100 out of 100 times, but was terminated each time because it yielded a singular matrix of second derivatives. In contrast, **A2**, converged to the right solution ($\widehat{\mu}_1 = 2.13$ and $\widehat{\mu}_2 = -2.67$) 98 out of 100 times. Table 5 presents a summary of the values to which algorithm **A2** converges, along with a summary of real times in seconds taken by **A2** to converge.

$Table\ 5$: A summary of the values yielded and real times taken by **A2**			
Algorithm	**A2**		
	$\widehat{\mu}_1$	$\widehat{\mu}_2$	real time (in seconds)
Max	2.19	-2.67	1.32
Mean	2.13	-2.67	1.15
Median	2.13	-2.67	1.16
Minimum	2.13	-2.67	0.95

3. Concluding remarks

The algorithm presented in this paper may be used for nonlinear or linear continuous optimization in possible presence of nonlinear and/or linear constraints. Special cases of such continuous optimization include both unconstrained and constrained maximum likelihood estimation. The algorithm works well in cases where the Newton-Raphson algorithm works well. This was illustrated through example 2.2. Moreoever, it works

well in cases where the Newton-Raphson doesn't, as was illustrated through example 2.3. Moreover, the examples also illustrated that the algorithm is quite fast in real time.

References

[1] Luenberger, D. G. (2004). Linear and Nonlinear Programming. Kluwer: Boston.

[2] Billingsley, P. (1995). Probability and Measure. Wiley: New York.