Optimization Methods I.

Newton-Raphson and others.

Anatoli Iambartsev
IME-USP

Motivation example. [FCN]

Consider the beta distribution. Suppose that $X_1, \ldots, X_n \sim B(p, q)$ and i.i.d. The density is

$$f(x; p, q) = \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} x^{p-1}(1-x)^{q-1}, \quad p, q > 0, x \in (0, 1),$$

where $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ is gamma function. Note that the uniform distribution is the particular case of gamma distribution with $p = q = 1$. Here

$$\mathbb{E}(X_i) = \frac{p}{p+q} \quad \text{and} \quad \mathbb{V}ar(X_i) = \frac{pq}{(p+q)^2(p+q+1)}.$$

Motivation example. [FCN]

Likelihood function for gamma distribution:
let $x = (x_1, \ldots, x_n)$

$$\begin{aligned} L \equiv L(p, q; x) &= \prod_{i=1}^{n} \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)} x_i^{p-1}(1-x_i)^{q-1} \\ &= \Big(\frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)}\Big)^n \prod_{i=1}^{n} x_i^{p-1}(1-x_i)^{q-1}. \end{aligned}$$

Log-likelihood function for gamma distribution:

$$\begin{aligned} \ell \equiv \ell(p, q; x) &= n \ln\big(\Gamma(p+q)\big) - n \ln\big(\Gamma(p)\big) - n \ln\big(\Gamma(q)\big) \\ &+ (p-1)\sum_{i=1}^{n} \ln(x_i) + (q-1)\sum_{i=1}^{n} \ln(1-x_i). \end{aligned}$$

Motivation example. [FCN]

In order to find the maximum likelihood estimator we have to resolve the following system of equations

$$\frac{\partial \ell}{\partial p} = 0 \quad \text{and} \quad \frac{\partial \ell}{\partial q} = 0.$$

We have no "closed" solution of the system. Thus we need to maximize the log-likelihood function numerically (no-linear optimization problem).

Question: how to find the maximum numerically?

Optimization.

Optimization problems:

(i) To find an extreme points of a function $h(\theta)$ in a domain $\theta \in \Theta$.

(ii) To find a solution (solutions) of an equation $g(\theta) = 0$ in a domain $\theta \in \Theta$.

Two type of problem can be considered as equivalent:

$(i) \rightarrow (ii)$ Reformulate the problem (ii) in the form of (i) by choosing $h(\theta) = g^2(\theta)$.

$(ii) \rightarrow (i)$ Reformulate the problem (i) in the form of (ii) by choosing $g(\theta) = \frac{dh(\theta)}{d\theta}$.

Optimization.

- Continuous versus Discrete Optimization
- Constrained versus Unconstrained Optimization
- Global versus Local Optimization
- Stochastic versus Deterministic Optimization

Optimization. Line search. [FCN]

Ideia. To find a maximum of a target function using *iterative scheme*.

Start at some inicial θ_0. If after n iterations θ_n is still not optimal value, calculate *directional vector* Δ_n and *step-length* λ_n and calculate the next value

$$\theta_{n+1} = \theta_n + \lambda_n \Delta_n.$$

This is our *general iterative scheme*.

[NW] call this strategy as *line search*. Note that for a given θ_n and direction Δ_n the method need a *secondary optimization* in order to find an optimal value of the step-length λ_n.

Optimization. Line search. [FCN]

$$\theta_{n+1} = \theta_n + \lambda_n \Delta_n.$$

[NW] call this strategy as *line search*. Note that for a given θ_n and direction Δ_n the method need a *secondary optimization* in order to find an optimal value of the step-length λ_n.

Observation: If we add in the search strategy an optimal value of λ_n, it makes the search computationally hard. Thus, the secondary optimization problem in this step in general is substituted by *ad hoc rules*.

Optimization. Trust Region. [NW. Chapter 2.2]

"In the second algorithmic strategy, known as *trust region*, the information gathered about target function f is used to construct a *model function* m_n whose behavior near the current point x_n is similar to that of the actual objective function f. Because the model m_n may not be a good approximation of f when θ is far from θ_n, we restrict the search for minimizer of m_n to some region around θ_n. In other words we find the candidate step p by approximately solving the following subproblem:

$$\min_p m_n(\theta_n + p),$$

where $\theta_n + p$ lies inside the trust region."

Gradient methods. [FCN]

More common algorithms are *gradient methods*. Here

$$\Delta_n = W_n g_n,$$

where W_n is some positive definite matrix and $g_n \equiv g(\theta_n)$ is the *gradient* of the objective function F:

$$g_n = \bigtriangledown F(\theta_n) \equiv \left.\frac{\partial F}{\partial \theta}\right|_{\theta_n}.$$

(obs.: the vectors are column vectors)

Background of Gradient Methods. [FCN]

Consider the Taylor expansion of $F(\theta_{n+1})$ around the point corresponding $\lambda_n = 0$:

$$F(\theta_{n+1}) \equiv F(\theta_n + \lambda_n \Delta_n) \approx F(\theta_n) + \lambda_n \bigtriangledown F(\theta_n)^T \Delta_n.$$

Let $F_{n+1} \equiv F(\theta_{n+1})$ and $F_n \equiv F(\theta_n)$ then

$$F_{n+1} - F_n \approx \lambda_n \bigtriangledown F(\theta_n)^T \Delta_n \equiv \lambda_n g_n^T \Delta_n.$$

If $\Delta_n = W_n g_n$ (gradient methods), then

$$F_{n+1} - F_n \approx \lambda_n g_n^T W_n g_n.$$

If $g_n \neq 0$ and λ_n is sufficiently small, then $F_{n+1} - F_n$ must be positive.

[FCN, NW]. Gradient Methods. Steepest descent method.

Different choices of W provide different methods.

Steepest descent method based on the following choice of W_n:

$$W_n = I,$$

where I is the identity matrix (in this case $\Delta_n = g_n$) with the following choice of step-length

$$\lambda_n = -\frac{g_n^T g_n}{g_n^T H_n g_n},$$

where

$$H_n \equiv H(\theta_n) = \bigtriangledown^2 F(\theta_n) \equiv \left.\frac{\partial^2 F(\theta)}{\partial \theta^2}\right|_{\theta_n}.$$

(H_n is Hessian matrix)

[NW Chapter 3.3]. Steepest descent method.

The rationalization on the choice of step-length can be illustrated on the ideal case for this method – when the objective function is quadratic (in this case the line searches are exact). Suppose that

$$F(\theta) = b^T\theta - \frac{1}{2}\theta^T Q\theta,$$

where Q is some symmetric and positive definite matrix. The gradient is given by $\bigtriangledown F(\theta) = b - Q\theta$, and the maximizer $\theta*$ is the unique solution of the linear system $Q\theta = b$.

[NW Chapter 3.3]. Steepest descent method.

Let us compute the step length λ_n that maximizes $F(\theta_n + \lambda \bigtriangledown F(\theta_n))$. By differentiating

$$F(\theta_n + \lambda p_n) = b^T(\theta_n + \lambda p_n) - \frac{1}{2}(\theta_n + \lambda p_n)^T Q(\theta_n + \lambda p_n)$$

with respect to λ

$$\begin{aligned} & b^T p_n - \frac{1}{2}\theta_n^T Q p_n - \frac{1}{2} p_n^T Q \theta_n - \lambda p_n^T Q p_n \\ & = (p_n + Q\theta_n)^T p_n - \frac{1}{2}\theta_n^T Q p_n - \frac{1}{2} p_n^T Q \theta_n - \lambda p_n^T Q p_n \\ & = p_n^T p_n - \lambda p_n^T Q p_n = 0, \end{aligned}$$

we obtain

$$\lambda_n = \frac{p_n^T p_n}{p_n^T Q p_n} \equiv \frac{\bigtriangledown F(\theta_n)^T \bigtriangledown F(\theta_n)}{\bigtriangledown F(\theta_n)^T Q \bigtriangledown F(\theta_n)}.$$

[NW Chapter 3.3]. Steepest descent method.

Finally

$$\theta_{n+1} = \theta_n + \frac{\bigtriangledown F(\theta_n)^T \bigtriangledown F(\theta_n)}{\bigtriangledown F(\theta_n)^T Q \bigtriangledown F(\theta_n)} \bigtriangledown F(\theta_n).$$

Remembering that around a maximum the second order Taylor expansion the Hessian is negative definite, then we can choose $Q = -H_n$, thus

$$\theta_{n+1} = \theta_n - \frac{\bigtriangledown F(\theta_n)^T \bigtriangledown F(\theta_n)}{\bigtriangledown F(\theta_n)^T H_n \bigtriangledown F(\theta_n)} \bigtriangledown F(\theta_n).$$

[NW, Chapter 2.2, p. 23]. Newton-Raphson method.

It based on the second-order Taylor series approximation to $F(\theta_n + p)$, which is

$$F(\theta_n+p) \approx F(\theta_n)+p^T \bigtriangledown F(\theta_n)+\frac{1}{2}p^T \bigtriangledown^2 F(\theta_n)p =: m_n(p).$$

Assuming for the moment that $\bigtriangledown^2 F(\theta_n)$ is negative definite, we obtain the Newton direction by finding the vector p that maximizes $m_n(p)$. By simply setting the derivative of $m_n(p)$ to zero, we obtain the following explicit formula:

$$p_n = -\left(\bigtriangledown^2 F(\theta_n)\right)^{-1} \bigtriangledown F(\theta_n).$$

[NW, Chapter 2.2, p. 23]. Newton-Raphson method.

The Newton direction is

$$p_n = -\left(\bigtriangledown^2 F(\theta_n)\right)^{-1} \bigtriangledown F(\theta_n).$$

Supposing that $\lambda_n \equiv 1$, we obtain the following iterative formula

$$\theta_{n+1} = \theta_n - \left(\bigtriangledown^2 F(\theta_n)\right)^{-1} \bigtriangledown F(\theta_n).$$

Newton-Raphson method, observation:

The high rate of convergence of the method is achieved due to the fact that it is a second-order method. Thus, its iteration is much more labor-intensive than, for example, the iteration of gradient methods. Fortunately, on the basis of Newton's method, there exist so-called quasi-Newtonian methods that are only slightly inferior to Newton's method on convergence rate, and their iterations are just a little more laborious than the iterations of gradient methods.

[FCN] Quasi-Newton methods.

$$\theta_{n+1} = \theta_n + \lambda_n \Delta_n, \quad \Delta_n = W_n g_n, \quad g_n = \bigtriangledown F(\theta_n) \equiv \left.\frac{\partial F}{\partial \theta}\right|_{\theta_n}.$$

A wide class of efficient algorithms called *Quasi-Newton methods* are based on the following idea: the iteration of the matrix W_n follows the rule

$$W_{n+1} = W_n + E_n,$$

where E_n is a some negative (positive) definite matrix. Note that if W_0 is negative (positive) definite (in general $W_0 = I$ is used), then W_n are positive (negative) definite for any $n \geq 0$.

[FCN] Quasi-Newton methods.

$$\theta_{n+1} = \theta_n + \lambda_n \Delta_n, \quad \Delta_n = W_n g_n, \quad g_n = \bigtriangledown F(\theta_n) \equiv \left.\frac{\partial F}{\partial \theta}\right|_{\theta_n}.$$

A wide class of efficient algorithms called *Quasi-Newton methods* are based on the following idea: the iteration of the matrix W_n follows the rule

$$W_{n+1} = W_n + E_n,$$

where E_n is a some negative (positive) definite matrix. The two algorithms are known

a) DFP (Davidon-Fletcher-Powell) method;

b) BFGS (Broyden-Fletcher-Goldfarb-Shanno) methods.

[FCN] Quasi-Newton: Davidon-Fletcher-Powell method

Based on paper of C.G.Broyden published 1969.

Let $\delta_n = \lambda_n \Delta_n$ (or $\theta_{n+1} - \theta_n = \delta_n$) and let $\gamma_n = g_{n+1} - g_n$. The DFP algorithm uses the following recursion

$$W_{n+1} = W_n + \frac{\delta_n \delta_n^T}{\delta_n^T \gamma_n} + \frac{W_n \gamma_n \gamma_n^T W_n}{\gamma_n^T W_n \gamma_n}.$$

Note that the formula can be represented in a way

$$W_{n+1} = W_n + aa^T + bb^T = W_n + [a, b][a, b]^T.$$

A matrix with two column $[a, b]$ has a rank 2. Thus one say that the algorithm DFP has a **correction of rank two**.

[FCN] Quasi-Newton: Broyden-Fletcher-Goldfarb-Shanno method.

It is an algorithm of the so-called **correction of rank three**. Let

$$\nu_n = \gamma_n^T W_n \gamma_n.$$

The algorithm use the following recurrent formula

$$W_{n+1} = W_n + \frac{\delta_n \delta_n^T}{\delta_n^T \gamma_n} + \frac{W_n \gamma_n \gamma_n^T W_n}{\gamma_n^T W_n \gamma_n} - \nu_n d_n d_n^T.$$

where

$$d_n = \left(\frac{1}{\delta_n^T \gamma_n}\right)\delta_n - \left(\frac{1}{\gamma^T W_n \gamma_n}\right) W_n \gamma_n.$$

There are evidences that this method is more efficient then the method DFP.

[FCN] Quasi-Newton methods.

$$W_{n+1} = W_n + E_n,$$

where E_n is a some negative (positive) definite matrix. The objective of the quasi-Newton algorithms is to obtain a good approximation for the inverse of Hessian:

$$W_n \approx -H^{-1},$$

when n is large enough.

Attention: a final matrix W can be a not good approximation for $-H^{-1}$. Some authors suggest to restart an algorithm and execute some more iterations.

[FCN] Convergence criteria.

We say that there is a *convergence* when the iteration achieves a *stability*.

There are various *convergence criteria* that can be used. Lot of them are based on a relative variations of a function and/or parameters.

In the case of Newton's like algorithms we can stop an algorithm when

$$\left|g_{n+1}^T H_{n+1}^{-1} g_{n+1} - g_n^T H_n^{-1} g_n\right| < \varepsilon,$$

for some given small ε.

References.

[FCN] Francisco Cribari-Neto *Elementos de Estatstica Computacional.* Mini-course on 5th Workshop on Probabilistic and Statistical Mwthods, February 6-8, 2017, ICMC-USP, So Carlos, SP, Brazil.

[NW] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 1999.

[RC] Cristian P. Robert and George Casella. Introducing Monte Carlo Methods with R. Series "Use R!". Springer