Optimization: Part 1

Abhinav Anand, IIMB

2018/06/21

Background

Problems in finance and economics are often concerned with the behavior of agents who are considered to be utility maximizers. Utility functions are thought to be monotonic in their variables—more of a utility enhancing variable is better than less— and are thought to obey diminishing returns as the variables scale.[1] Additionally, real-life constraints ensure that variables are bounded. Hence optimization of functions under constraints forms an important discipline in the study of such subjects.

After linear optimization, quadratic optimization problems are the simplest since they can be captured in *quadratic forms* and can be represented via symmetric matrices with special properties. A distinguishing feature of quadratic optimization is the presence of first and second order conditions that characterize the nature of the extreme point.

Quadratic Optimization

In one dimension, $x \in \mathbb{R}$ the simplest quadratic objective functions can be $f(x) = \{x^2, -x^2\}$ with the first attaining a global minimum and the second a global maximum at $x = 0$.

```
x <- -500:500
y_1 <- x^2
y_2 <- -x^2
data_l <- cbind(x, y_1, y_2) %>%
  dplyr::as_tibble() %>% #wide format
```

---

[1]This utility function can assume a variety of forms. For example, if the 'agent' is a firm, its utility is its profit function; if it's a government, it could be some (aggregate) social welfare function etc.

```
tidyr::gather(.,
              y_1:y_2,
              key = 'y',
              value = 'f') #long format

ggplot(data_l, aes(x, f, color = y)) +
  geom_line() +
  geom_hline(yintercept = 0) +
  theme_minimal()
```
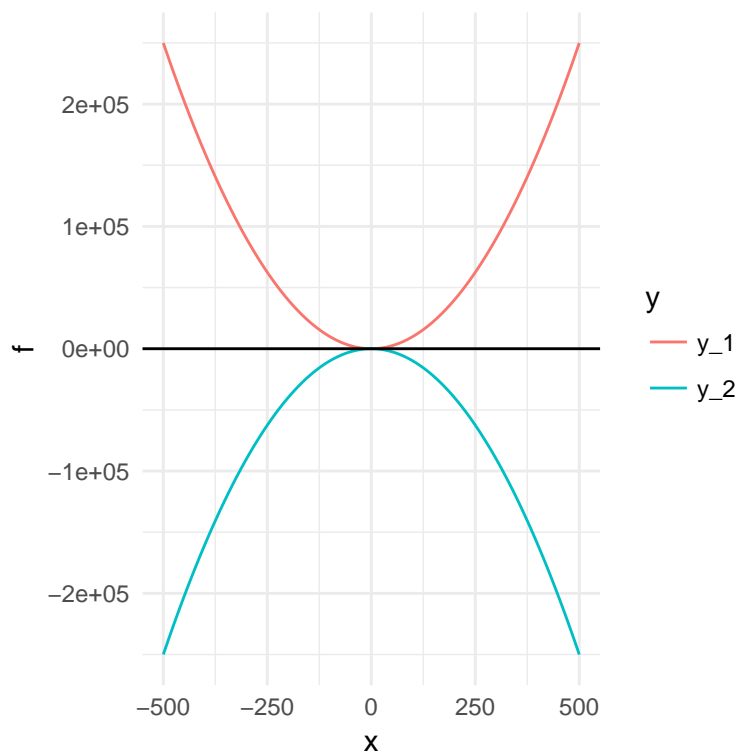

Quadratic Forms

For the case when $x = (x_1, x_2) \in \mathbb{R}^2$, a general quadratic form is the following:

$$Q(x) = a_{11}x_1^2 + a_{22}x_2^2 + 2a_{12}x_1x_2$$

which may be represented as a matrix:

$$Q(x) = [x_1 x_2] \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Similarly, when $x = (x_1, x_2, x_3) \in \mathbb{R}^3$, the form becomes

$$Q(x) = [x_1 x_2 x_3] \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

and in general, when $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$:

$$Q(x) = [x_1 \ldots x_n] \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & & \\ a_{1n} & \ldots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

It's easy to see that $Q(0) = 0$. If $a > 0, ax^2 > 0$ and we can call the quadratic form "positive definite". Similarly, if $a < 0, ax^2 < 0$ and we call the form "negative definite". This carries over when $x \in \mathbb{R}^2$, since $Q(x_1, x_2) = x_1^2 + x_2^2 > 0$ (positive definite) and $Q(x_1, x_2) = -x_1^2 - x_2^2 > 0$ (negative definite). Additionally for the case when $Q(x_1, x_2) = x_1^2 - x_2^2 > 0$, the quadratic form is *indefinite*.

In general, symmetric matrices are called positive (semi)definite, negative (semi)definite etc. according to the definiteness of the expression $Q(x) = x^\top A x$.[2] Additionally for symmetric matrices, the eigenvalues are all real numbers. A highly useful fact to know is that for positive definite matrices, all eigenvalues are positive; for negative definite matrices, all eigenvalues are negative; and for indefinite matrices, some eigenvalues are positive and some are negative.

Computation: Quadratic Programming

Unconstrained Optimization

While it's clear what a maximum is in one dimension, what should be its definition in two dimensions and beyond? The central idea remains the same: x^* is the maximum

---

[2]Hence if $\forall x \neq 0 \in \mathbb{R}^n, x^\top A x \geq 0$ the form is positive semidefinite and so on.

if there is no other x in the domain of $f(x)$ such that $f(x) \geq f(x^*)$. Note that this definition is general and independent of the dimension of the underlying space from where we pick x. The only difference in dimensions one and beyond is the structure of the domain; and while in one dimension it must take the form $x_L \leq x \leq x_H$, one needs such conditions for each component when $x \in \mathbb{R}^n : x_{L1} \leq x_1 \leq x_{H1}, \ldots, x_{Ln} \leq x_n \leq x_{Hn}$. With this interpretation in mind, we are ready to formulate the necessary and sufficient conditions for the optimal of an unconstrained optimization program.

First Order Conditions

When the dimension of the domain is 1, $x_L \leq x \leq x_H$, the first order condition stipulates that at the *critical point* the derivative be 0.[3] Whether this critical point is a (local) maximum or (local) minimum or neither, depends on the *second order condition* on the derivative of the derivative.

The example of $f(x) = x^2$ illustrates this idea.

```
(plot_x_square <- ggplot(data = dplyr::as_tibble(cbind(x, y_1)),
        mapping = aes(x, y_1)) +
  geom_line() +
  theme_minimal())
```


---
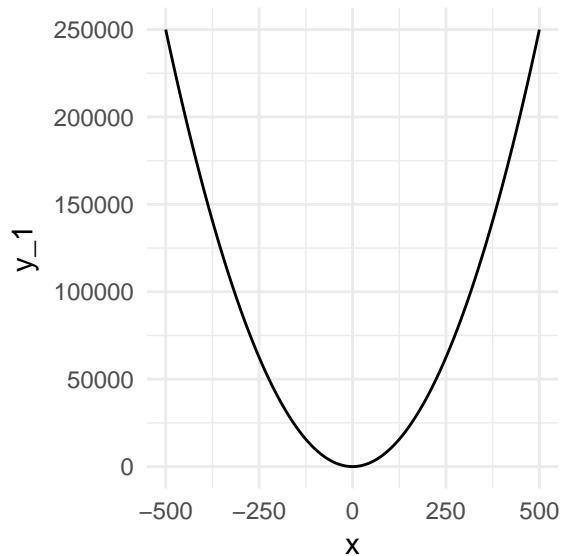
[3]Why must this be so? Can you see the case for the plot $f(x) = x^2$ and reason why?

End points or corner points are generally not considered critical points though they can contain extreme points. (For the plot $f(x) = x^2$ what is/are maximal points if $x \in [l, b]$?) Conventionally, critical points are thought to be points interior to the domain.

Functions can be separated into classes on the basis of how differentiable they are (the differentiability class). For example, all functions that are continuous form the C^0 class; those that are differentiable and their derivatives also continuous form the class C^1; those whose second derivatives are continuous form the class C^2 and so on.[4]

Theorem: If $x^* \in \mathbb{R}^n$ is an interior point in the domain of the real-valued function $F \in C^1$, then it is a critical point when

$$\frac{\partial F}{\partial x_i}(x^*) = 0, i \in \{1, 2, \ldots, n\}$$

Second Order Conditions

Critical points are interior points at which all first partial derivatives are 0. However, this in itself is not enough for us to know if the critical point is a maximum, a minimum or neither. For this, we need to consider its second derivatives.

For the function $F \in C^2, x \in U \subset \mathbb{R}^n$, ($U$ is open in $\mathbb{R}^n$) the *Hessian* is the following second derivative matrix:

$$D^2 F(x^*) = \begin{bmatrix} \frac{\partial F}{\partial x_1^2}(x^*) & \cdots & \frac{\partial F}{\partial x_n \partial x_1}(x^*) \\ \vdots & \ddots & \vdots \\ \frac{\partial F}{\partial x_1 \partial x_n}(x^*) & \cdots & \frac{\partial F}{\partial x_n^2}(x^*) \end{bmatrix}$$

Since for the function $F \in C^2$, the cross-partials $\frac{\partial F}{\partial x_i \partial x_j} = \frac{\partial F}{\partial x_j \partial x_i}$, the Hessian is symmetric.

The second order conditions are based on the Hessian. Just as in the one dimensional case, where there occurs a maximum at a critical point if the second derivative is

---

[4]A function is "smooth" if all derivatives are continous: $f \in C^\infty$; and analytical $f \in C^\omega$ if it is smooth *and* its Taylor approximation converges to its function value at all points in the domain.

negative; there occurs a maximum at the critical point $x^* \in U \subset \mathbb{R}^n$ if the Hessian is *negative definite.*

Theorem: For the function $F \in C^2, x \in U \subset \mathbb{R}^n$ with critical point x^*:

1. If the Hessian $D^2 F(x^*)$ is negative definite, then x^* is a local maximum.
2. If the Hessian $D^2 F(x^*)$ is positive definite, then x^* is a local minimum.
3. If the Hessian $D^2 F(x^*)$ is indefinite, then x^* is neither. (x^* is a "saddle point".)

Convex Functions and Global Minima

While the conditions above classify critical points as maximal, minimal or saddle; these are only local optima. However if we consider only a special class of functions called *convex*, we can guarantee that the local minimum will also be the global minimum!

A simple example of a convex function is $f(x) = x^2$. It has one minimum at $x = 0$ which is also its global minimum. The added wrinkle is that the domain of a convex function must be a *convex* set.

Convex Set

A set $U \subset \mathbb{R}^n$ is convex if for two points in it, the whole line connecting the two points must also fully lie in U. More formally, a set U is convex if $\forall t \in [0, 1]$

$$x, y \in U \Rightarrow tx + (1 - t)y \in U$$

(Can you see this geometrically?)

Convex Function

Now, we can define a convex *function* whose domain is a convex *set* by stipulating that the image of the line connecting x to y lie below the line itself. Hence a function F is convex if its domain $U \in \mathbb{R}^n$ is convex; and for all $x, y \in U$ and $\forall t \in [0, 1]$

$$tF(x) + (1 - t)F(y) \geq F(tx + (1 - t)y)$$

We can check if this condition is satisfied for our one dimensional convex function $f(x) = x^2$.

```
plot_x_square +
  geom_segment(aes(x = 0,
                   y = 0,
                   xend = 400,
                   yend = 400^2
                   ),
                   linetype = "dotdash",
                   color = "red"
                   )
```
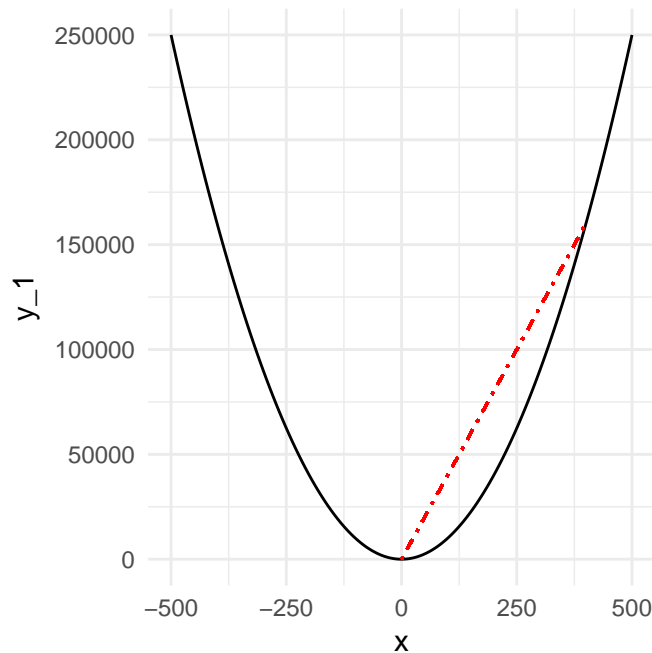

Concave Function

A *concave* function is simply the negative of a convex function. In one dimension, $-x^2, \log(x), \sqrt{x}$ etc. are some commonly known concave functions. For these, local maxima are global maxima. Note that just like the convex functions, domains of concave functions must be convex too.

Theorem: If $F : U \to \mathbb{R}$, (where $U \subset \mathbb{R}^n$ is convex and open) $F \in C^2$ and convex then

1. $D^2F(x)$ is positive semidefinite for $x \in U$
2. $F(y) - F(x) \geq DF(x) \cdot (y - x)$ for $x, y \in U$
3. If for $x^* \in U, DF(x^*) = 0$ then x^* is the global minimum

Computation: Unconstrained Optimization

There are several special packages written in `R` for solving optimization problems. A core set of packages is included with the installation of `R`, with more than 12,500 additional packages (as of May 2018) available at the "Comprehensive R Archive Network" (CRAN). Additionally, packages related to a topic, say, Econometrics or Optimization or Differential Equations etc. can be found by looking them up in CRAN Task Views: https://cran.r-project.org/web/views/.

We quote here from CRAN Task Views: Optimization

> For one-dimensional unconstrained function optimization there is `optimize()` which searches an interval for a minimum or maximum. Function `optim()` provides an implementation of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, bounded BFGS, conjugate gradient (CG), Nelder-Mead, and simulated annealing (SANN) optimization methods. It utilizes gradients, if provided, for faster convergence. Typically it is used for unconstrained optimization but includes an option for box-constrained optimization.

> Additionally, for minimizing a function subject to linear inequality constraints, stats contains the routine `constrOptim()`. Then there is `nlm` which is used for solving nonlinear unconstrained minimization problems.

Consider the function $f(x) = 2(x_1 - 1)^2 + 5(x_2 - 3)^2 + 10$

```
func_1_x <- function(x)
{
  2*(x[1]-1)^2 + 5*(x[2]-3)^2 + 10
}
```
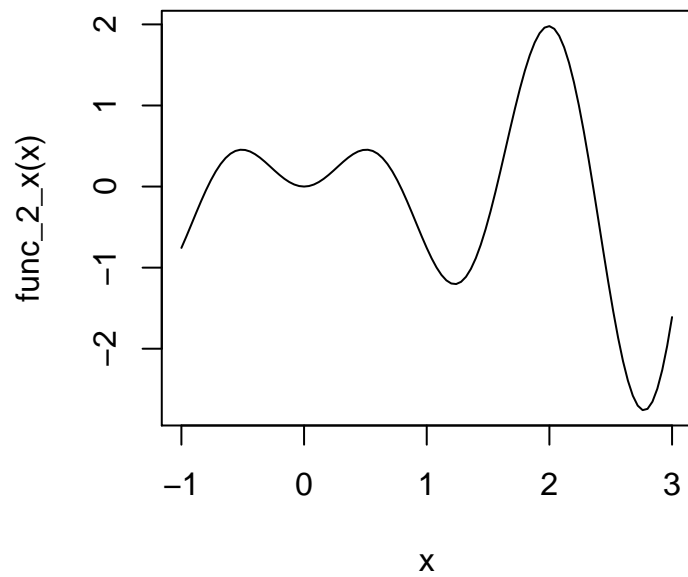
```r
optim_func_1_x <- optim(c(1,1), func_1_x) #x=(x_1,x_2)=c(1,1)
print(optim_func_1_x)
```

```
## $par
## [1] 1.000168 3.000232
##
## $value
## [1] 10
##
## $counts
## function gradient
##       75      NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```
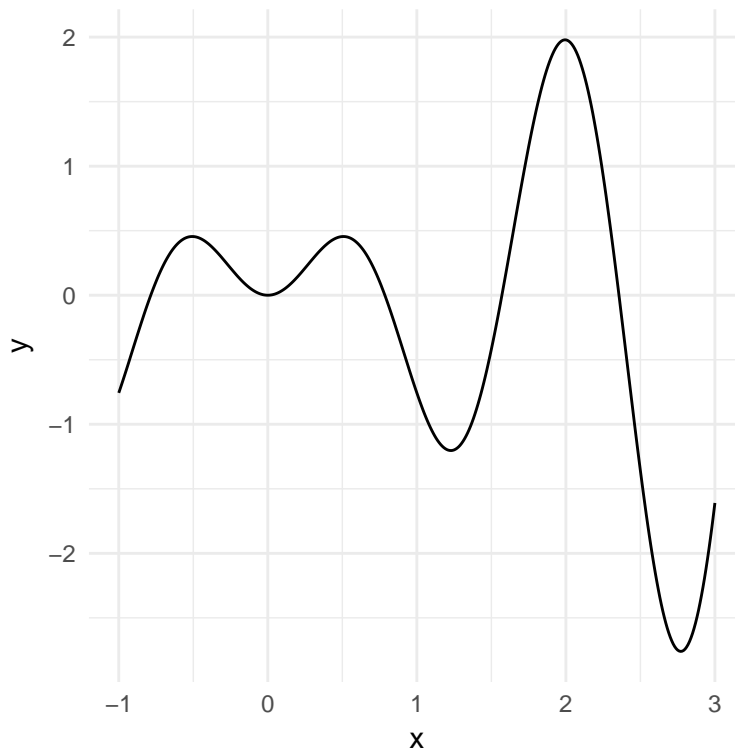
```r
# Another function for optimization
func_2_x <- function(x) {x*sin(4*x)}


curve(func_2_x, -1, 3) #note base R graphics function curve()
```

```r
# Via ggplot()
x <- seq(-1, 3, 0.01)
y <- func_2_x(x)
data_plot_gg <- cbind(x, y) %>% tibble::as_tibble()

ggplot(data_plot_gg, aes(x, y)) +
  geom_line() +
  theme_minimal()
```

```
optimize(func_2_x, c(0, 3))
```

```
## $minimum
## [1] 1.228297
##
## $objective
## [1] -1.203617
```

Application: Least Squares, Weighted Least Squares

Application: Maximum Likelihood Estimation