IEOR 4004: Optimization Models and methods

Lecture : Nonlinear Programming[1]

In linear programming, we considered the problem of maximizing/minimizing a linear function of decision variables $x_1, \ldots, x_n$, given linear inequality and/or equality constraints:

$$\begin{array}{ll} \min & \sum_{j=1}^n c_j x_j \\ & \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, \ldots, m \\ & \sum_{j=1}^n a'_{ij} x_j = b_i, \quad i = 1, \ldots, m' \end{array}$$

A general nonlinear program with n variables and m constraints can be stated as

$$\begin{array}{ll} \min & f(x_1, x_2, \ldots, x_n) \\ & g_i(x_1, \ldots, x_n) \leq b_i, \quad i = 1, \ldots, m \\ & h_i(x_1, \ldots, x_n) = b_i, \quad i = 1, \ldots, m' \end{array}$$

where f, g_i are arbitrary nonlinear functions of $x_1, \ldots, x_n$. Or, in a more concise form:

$$\begin{array}{ll} \min & f(x) \\ & g_i(x) \leq b_i, \quad i = 1, \ldots, m \\ & h_i(x) = b_i, \quad i = 1, \ldots, m' \end{array} \tag{1}$$

where x is the vector (typically column vector) of decision variables $x_1, \ldots, x_n$.

1 Example Applications

Mean-Variance Portfolio Selection Consider the following introductory case. An investor has \$5000 and two potential investments. Let x_j for $j = 1$ and $j = 2$ denote the dollar allocation to investment j. From historical data, investments 1 and 2 have an expected annual return of 20 and 16 percent, respectively. Also, the total risk involved with investments 1 and 2, as measured by the variance of total return, is given by $2x_1^2 + x_2^2 + (x_1 + x_2)^2$, so that risk increases with total investment and with the amount of each individual investment. The investor would like to maximize his expected return and at the same time minimize his risk. Clearly, both of these objectives cannot, in general, be satisfied simultaneously. There are several possible approaches. For example, the goal can be to minimize risk subject to a constraint imposing a lower bound on expected return. Alternatively, expected return and risk can be combined in an objective function, to give the model:

$$\text{Maximize } f(x) = 20x_1 + 16x_2 - \theta[2x_1^2 + x_2^2 + (x_1 + x_2)^2] \tag{2a}$$

$$\text{subject to:} \tag{2b}$$

$$g_1(x) = x_1 + x_2 \leq 5, \tag{2c}$$

$$x_1 \geq 0,\ x_2 \geq 0. \tag{2d}$$

The non-negative constant θ reflects his tradeoff between risk and return. If $\theta = 0$, the model is a linear program, and he will invest completely in the investment with greatest expected return. For very large θ, the objective contribution due to expected return becomes negligible and one is essentially minimizing risk.

In practice there is an alternative formulation that is more common. Formulation (2) is on a "dollar" basis, i.e. the return term in the objective is measured in dollars. Instead in practice one focuses on the portfolio return as a

percentage quantity and the problem is stated as a minimization. Let y_j $(j = 1, 2)$ be the percentage invested on asset j. The alternate formulation is

$$\text{Minimize } -20y_1 - 16x_y + \lambda[2y_1^2 + y_2^2 + (y_1 + y_2)^2] \tag{3a}$$

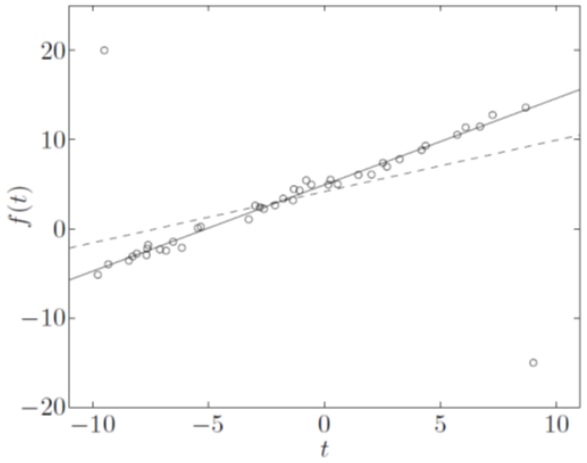
$$\text{subject to:} \tag{3b}$$

$$y_1 + y_2 \leq 1, \tag{3c}$$

$$y_1 \geq 0,\ y_2 \geq 0. \tag{3d}$$

The correspondence between (2) and (3) is that $x_j = 5y_j$ for $j = 1, 2$, and $\theta = 25\lambda$. Formulation (3) is more common because it allows managers to think in terms of percentage points, and so the portfolio can be scaled (in principle) to any overall budget amount. Frequently, (3c) may be replaced by $y_1 + y_2 = 1$ (full investment).

This is a very simple example of *porfolio optimization*. Realistic examples involve thousands of assets and budgets in the billions. The classical reference is *Portfolio Selection: Efficient Diversification of Investments,* by Harry Markowitz. A more modern reference is, e.g. *Modern Portfolio Theory and Investment Analysis,* by Elton, Gruber, Brown and Goetzmann, ISBN 0471238546.



Linear regression We are given vectors $x_1, x_2, \ldots, x_N$ all in $\mathbb{R}^n$. (E.g., x_i represents attributes like Age, Gender, Race, Marital status, Education level of the i^{th} person). For each x_i, we are given a scalar value y_i (E.g., salary level for the i^{th} person). We want to come up with an "approximate" **linear** model for the data. That is, we want to find $w \in \mathbb{R}^n$ and $z \in \mathbb{R}$ so that

$$y_i \approx w^T x_i + z, \quad 1 \leq i \leq N$$

Here decision variables are w, z.
(E.g., with such a linear model, we can predict the salary level, given a new person's attributes)

To measure the accuracy of this model, we can use a non-decreasing function $\ell(\cdot)$ of the error $(y_i - w^T x_i - z)$ for each i. A general linear regression problem is then stated as:

$$\min_{w,z} \sum_i \ell(y_i - w^T x_i - z)$$

This is an unconstrained, but non-linear optimization problem. Different linear regression approaches differ in the loss function $\ell(\cdot)$ used, some common choices are:

- Square loss (Quadratic loss function) $\ell(u) = u^2$:

$$\min_{w,z} \sum_i (y_i - w^T x_i - z)^2$$

- Absolute or Laplace loss or L_1 loss $\ell(u) = |u|$:

$$\min_{w,z} \sum_i |y_i - w^T x_i - z|$$

  This is an unconstrained non-linear problem, which can infact be converted to a constrained linear program, as we saw in the first lecture.

- Deadzone linear (ignore errors in a small region near 0)

$$\ell(u) = \begin{cases} 0, & |u| < 1 \\ 2|u| - 1, & otherwise \end{cases}$$

- Huber penalty

$$\ell(u) = \begin{cases} u^2, & |u| < 1 \\ 2|u| - 1, & otherwise \end{cases}$$

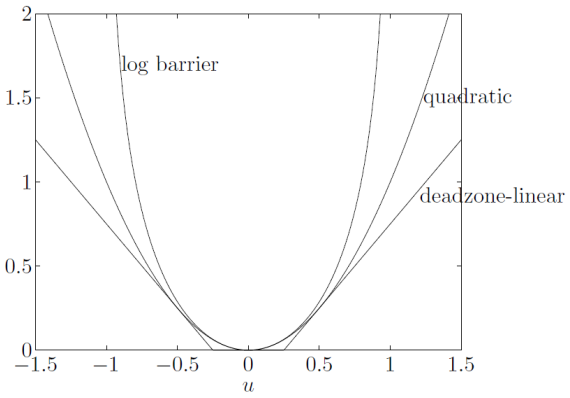
  Smaller penalty for smaller errors.

- Log-barrier

$$\ell(u) = -\log\left(1 - \frac{u^2}{M}\right)$$

  for some constant M, when $u \in (-M, M)$. (Heavy penalty for errors near the boundary of the error range)

The figure below plots these different loss functions for $u \in [-0.5, 0.5]$.

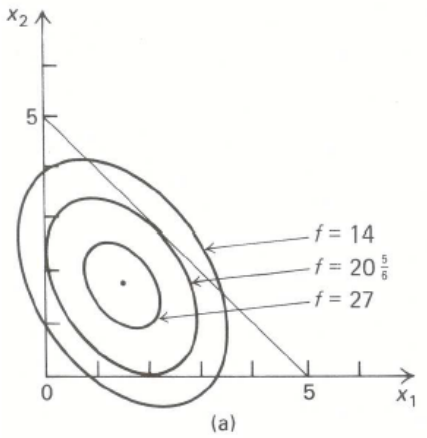


The shape of the penalty function has a large effect on the solution of the problem. Roughly speaking, $\ell(u)$ is a measure of our dislike of a residual of value u. If $\ell(u)$ is very small (or even zero) for small values of u, it means we care very little (or not at all) if residuals have these values. If $\ell(u)$ grows rapidly as u becomes large, it means we have a strong dislike for large residuals; if $\ell(u)$ becomes infinite outside some interval, it means that residuals outside the interval are unacceptable. This simple interpretation gives insight into the solution of a penalty function approximation problem, as well as guidelines for choosing a penalty function.

2 Characterizing the optimal solution

2.1 Interior vs. Boundary solutions.

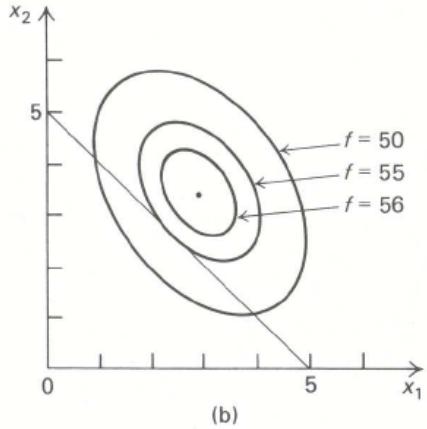
Geometrically, nonlinear programs can behave much differently from linear programs, even for problems with linear constraints. In Figure 1, the portfolio-selection example from the last section has been plotted for several values of the tradeoff parameter θ. For each fixed value of θ, contours of constant objective values are concentric ellipses.



$\theta = \frac{8}{5}$

Optimum $x_1 = \frac{3}{2}$, $x_2 = \frac{7}{4}$

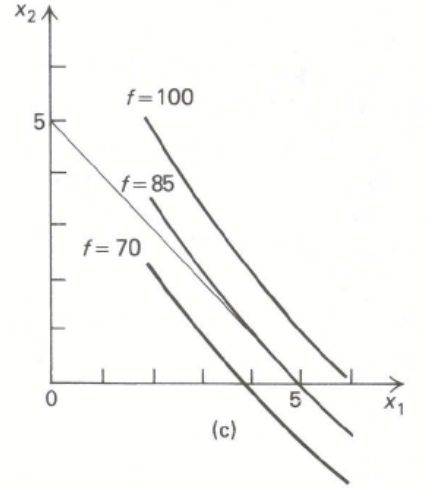
(Unconstrained optimum $x_1 = \frac{3}{2}$, $x_2 = \frac{7}{4}$)



$\theta = \frac{4}{5}$

Optimum $x_1 = 2.5$, $x_2 = 2.5$

(Unconstrained optimum $x_1 = 3$, $x_2 = 3.5$)



$\theta = \frac{1}{5}$

Optimum $x_1 = 5$, $x_2 = 0$

(Unconstrained optimum $x_1 = 12$, $x_2 = 14$)

Figure 1: Portfolio-selection example for various values of θ. Lines are contours of constant objective values. (Figure taken from [1], Figure 13.1)

As Figure 1 shows, the optimal solution can occur:

a) at an interior point of the feasible region;

b) on the boundary of the feasible region, which is not an extreme point; or

c) at an extreme point of the feasible region.

2.2 Local vs. global optimal

Definition 2.1 (Optimal solution)**.** *An optimal solution to the nonlinear program* (1) *is defined as a feasible point* x *such that* $f(x) \geq f(y)$ *for all feasible* y*.*

In a linear program, given any sub-optimal feasible point y, one can always find a point arbitrarily close to y which is better than y, i.e., which improves the objective value. This is not always the case for nonlinear programs. In fact, there can exist a sub-optimal point that is "locally optimal", i.e., its objective value is at least as good as any other feasible point in close vicinity. Figure 2 illustrates such a situation,

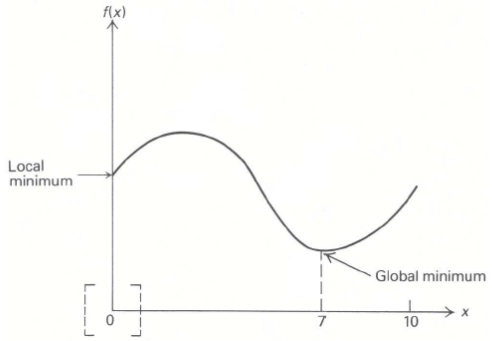


Figure 2: Local vs. Global minimum (Non-convex Non-concave function)

Definition 2.2 (Locally optimal solution)**.** *A local maximum is defined as a feasible point* x *such that* $f(x) \geq f(y)$ *for every feasible point* y *which is within* ϵ *distance of* y *for some (possibly very small)* $\epsilon > 0$*. Local minimum is defined analogously.*

To draw a contrast with local optimal solutions, optimal solutions for nonlinear program are also referred to as "global optimal" solutions. Clearly, every global optimal is also a locally optimal solution, but not the other way around.

The concept of a local maximum is extremely important. Most general-purpose nonlinear programming procedures are near-sighted and can do no better than determine local maxima. We should point out that, since every global maximum is also a local maximum, the overall optimization problem can be viewed as seeking the best local maxima. Under certain circumstances, local maxima and minima are known to be global. Whenever a function "curves upward" as in Figure 3(a), a local minimum will be global. These functions are called convex. Whenever a function "curves downward" as in Figure 3(b) a local maximum will be a global maximum. These functions are called concave. For this reason we usually wish to minimize convex functions and maximize concave functions. These observations are formalized below.

3 Convex Optimization

A convex program is an optimization problem where objective function is convex/concave (convex in minimization problem, concave in maximization problems) and feasibility set is convex. First, we formally define convex/concave functions, and convex sets.

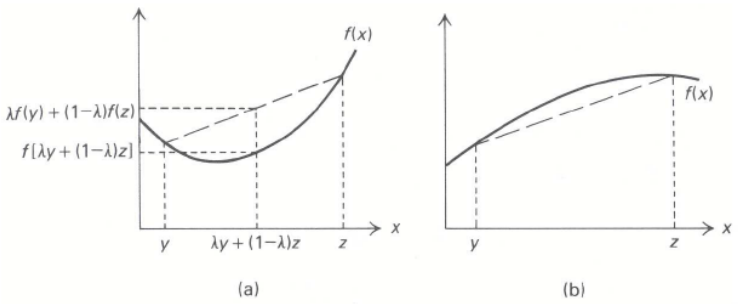


Figure 3: Convex and Concave functions

Definition 3.1. *A function $f(x)$ is called convex if, for every y and z and every $0 \leq \lambda \leq 1$,*

$$f(\lambda y + (1-\lambda)z) \leq \lambda f(y) + (1-\lambda)f(z).$$

An intuitive way to view a convex function is to note that linear interpolation overestimates its values. That is, for any points y and z, the line segment joining $f(y)$ and $f(z)$ lies above the function (see Figure 3). More intuitively, convex functions are "bathtub like" and hold water. *Concave functions are simply the negative of convex functions.* In this case, linear interpolation underestimates the function. The definition above is altered by reversing the direction of the inequality.

Definition 3.2. *A function $f(x)$ is called concave if, for every y and z and every $0 \leq \lambda \leq 1$,*

$$f(\lambda y + (1-\lambda)z) \geq \lambda f(y) + (1-\lambda)f(z).$$

A notion intimately related to convex and concave functions is that of a convex set. These sets are "fat", in the sense that, whenever y and z are contained in the set, every point on the line segment joining these points is also in the set.

Definition 3.3. *A set of points C is called convex if, for every $0 \leq \lambda \leq 1$, $\lambda x + (1-\lambda)y$ is contained in C whenever x and y are contained in C.*

Given these definitions, we can define a convex program.

Convex Program: A general convex program (minimization) is stated as:

$$\begin{aligned} \min \quad & f(x) \\ \text{subject to:} \quad & x \in C \end{aligned} \tag{4}$$
$$\tag{5}$$

where x is a vector of n decision variables $x_1, \ldots, x_n$, f is a *convex* function in x and C is a convex set. For example, the set

$$C = \{x : g_i(x_1, \ldots, x_n) \leq b_i, i = 1, \ldots, m, h_i(x_1, \ldots, x_n) = b_i, i = 1, \ldots, p\}$$

is convex if g_i are convex functions, and h_i are linear functions.

Equivalently, it can be defined as the maximization problem:

$$\begin{aligned} \max \quad & f(x) \\ \text{subject to:} \quad & x \in C \end{aligned} \tag{6}$$
$$\tag{7}$$

where x is a vector of n decision variables $x_1, \ldots, x_n$, f is a *concave* function in x and C is a convex set.

Following observation makes convex programming particularly useful.

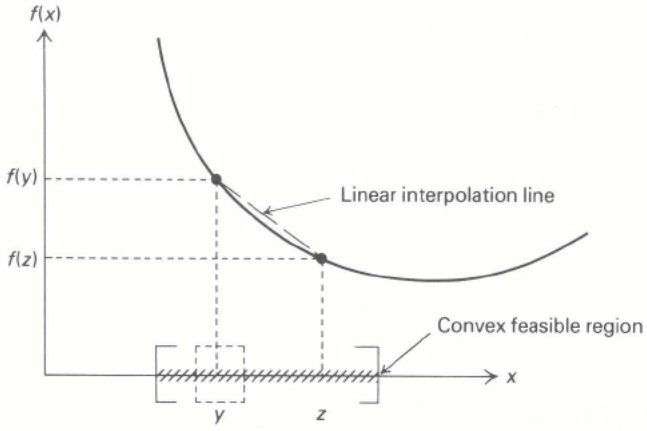


Figure 4: Local minima are global minima for convex function

Theorem 3.4. *A local minimum (maximum) for a convex (concave) function on a convex feasible region is also a global minimum (maximum).*

We can establish this property easily by reference to Figure 4. The argument is for convex functions; the concave case is handled similarly. Suppose that y is a local minimum. If y is not a global minimum, then, by definition, there is a feasible point z with $f(z) < f(y)$. But then if f is convex, the function must lie on or below the dashed linear interpolation line. Thus, in any box about y, there must be an x on the line segment joining y and z, with $f(x) < f(y)$. Since the feasible region is convex, this x is feasible and we have contradicted the hypothesis that y is a local minimum. Consequently, no such point z can exist and any local minimum such as y must be a global minimum.

Next, we study some equivalent, but often easier to check, conditions for convexity when the function is differentiable, and twice differentiable, respectively.

3.1 Differentiable convex functions

Definition 3.5. *A differentiable function $f : \Omega \to \mathbb{R}$ is convex iff*

$$f(z) + \nabla f(z)^T (x - z) \leq f(x)$$

for all $x, z \in \Omega$, where Ω is the domain of function f.

We can explain the reason why this definition corresponds to convexity in two steps. First let's consider a one-dimensional example $f(x)$ where x is a scalar. Then convexity means that the curve representing the function $f(x)$ is entirely "above" the *tangent* at any x-coordinate value z. Consider Figure 5.

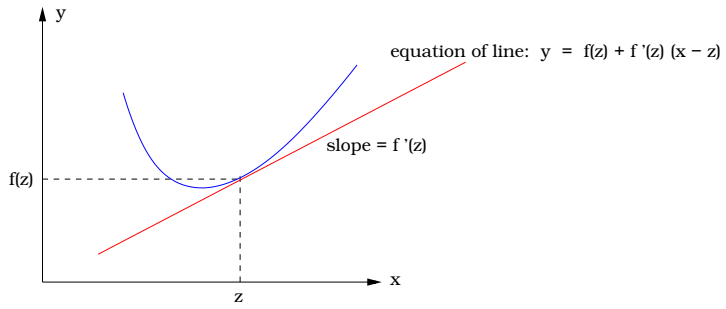


Figure 5: A differentiable convex function of a scalar

In this figure the tangent at the point $(z, f(z))$ is given by the equation $y = f(z) + f'(z)(x - z)$ using basic concepts of Calculus. So in this case we see that a differentiable function f on a scalar x is convex if and only if

$$f(x) \;\geq\; f(z) + f'(z)(x - z) \tag{8}$$

for every z and x. We want to understand how Definition 3.5 corresponds to this simple geometric insight.

To do so, consider a function $f(x)$ where x is n-dimensional. If f is convex then we should observe convexity ("upwards curvature") if we restrict x to a one-dimensional set. To that effect, let z be any (n-dimensional) point and let u be a unit n-dimensional vector. Then the set of points of the form $z + tu$, where t is a scalar (positive or negative) describe a straight-line in n-dimensional space that goes through the point z. The function of interest is $f(z + tu)$ which describes the value of f as t varies, for this fixed choice of z and u. Let's call this function $\mathbf{h(t)}$. So we have that $h(t)$ is convex. And if we apply 8 we should then have that

$$h(t) \;\geq\; h(0) + h'(0)t, \quad \text{for any } t. \tag{9}$$

There remains the task of computing $h(0)$ and $h'(0)$ in terms of f and u. First, $h(0) = f(z)$. And the value $h'(0)$ is called the *directional derivative* of f, at z, in the direction of u. It can be shown that $h'(0)$ equals

$$\nabla f(z)^T u. \tag{10}$$

Before we see the impact of this fact in terms of Definition 3.5, consider the example

$$f(x) \;=\; f(x_1, x_2) \;=\; 2^{x_1} + x_2^4$$

which can be shown to be convex. Let $z = (1, 1)^T$. Then $f(z) = 3$ and $\nabla f(z) = (2 \ln 2, 4)^T$.

- Consider first the unit vector $u = (1, 0)^T$. Then

  $$h(t) = f(z + tu) = f(z_1 + tu_1, z_2 + tu_2) = f(1 + t, 1) = 2^{1+t} + 1.$$

  The derivative of this function of t equals $2^{1+t} \ln 2$. So $h'(0) = 2 \ln 2$. On the other hand, if we use formula (10) we get

  $$2 \ln 2 u_1 + 4u_2 \;=\; 2 \ln 2.$$

  This agrees with $h'(0)$.

- Now consider the unit vector $u = (0, 1)^T$. Then

  $$h(t) = f(z + tu) = f(z_1 + tu_1, z_2 + tu_2) = f(1, 1 + t) = 2 + (1 + t)^4.$$

  The derivative of this function of t equals $4(1 + t)^3$. So $h'(0) = 4$. On the other hand, if we use formula (10) we get

  $$2 \ln 2 u_1 + 4u_2 \;=\; 4.$$

  Again this agrees with $h'(0)$.

- Finally consider $u = (1/\sqrt{2}, 1/\sqrt{2})^T$. Then

$$h(t) = f(z + tu) = f(z_1 + tu_1, z_2 + tu_2) = f(1 + t/\sqrt{2}, 1 + t/\sqrt{2}) = 2^{1+t/\sqrt{2}} + (1 + t/\sqrt{2})^4.$$

  The derivative of this function of t equals $2 \ln 2/\sqrt{2}\, 2^{t/\sqrt{2}} + 4/\sqrt{2}(1 + t/\sqrt{2})^3$. So $h'(0) = 2 \ln 2/\sqrt{2} + 4/\sqrt{2}$. And if we use formula (10) we get

$$2 \ln 2/\sqrt{2} + 4/\sqrt{2}$$

  which once again yields $h'(0)$.

Let's accept as given that $h'(0) = \nabla f(z)^T u$. Then applying 8 to the convex function $h(t)$ at $t = 0$ we get

$$f(z + tu) = h(t) \geq h(0) + h'(0)t \;=\; f(z) + \nabla f(z)^T ut = f(z) + \nabla f(z)^T (x - z)$$

which verifies Definition 3.5.

Likewise, we have

Definition 3.6. *A differentiable function $f : \Omega \to \mathbb{R}$ is concave iff*

$$f(z) + \nabla f(z)^T (x - z) \geq f(x)$$

for all $x, z \in \Omega$, where Ω is the domain of function f.

3.2 Twice differentiable convex functions

For a function of 1 variable, it is easy to see the condition $f''(x) \geq 0$ for convexity, which means that the derivative is non-decreasing. It can be interpreted geometrically as the requirement that the graph of the function have positive (upward) curvature at every x. The general condition for twice differentiable functions is stated as follows:

Definition 3.7. *A twice differentiable function f is convex if and only if its Hessian is positive semidefinite: for all $x \in \Omega$,*

$$\nabla^2 f(x) \succeq 0.$$

Here, the Hessian $H = \nabla^2 f(x)$ is an $n \times n$ matrix, with entry $H_{ij} = \frac{d^2}{dx_i dx_j} f(x)$.

Recall that a matrix H is positive semidefinite if and only $x^T Hx \geq 0$ for all x. Similarly,

Definition 3.8. *A twice differentiable function f is concave if and only if its Hessian is negative semidefinite: for all $x \in \Omega$,*

$$\nabla^2 f(x) \preceq 0.$$

Here, the Hessian $H = \nabla^2 f(x)$ is an $n \times n$ matrix, with entry $H_{ij} = \frac{d^2}{dx_j dx_j} f(x)$.

All linear functions are twice differentiable with $\mathbf{0}$ second derivative, hence they are trivially convex and concave. Another popular example of Twice differentiable functions is Quadratic functions.

Quadratic functions. Consider the quadratic function $f : \mathbb{R}^n \to \mathbb{R}$, with given by

$$f(x) = (1/2)x^T Px + q^T x + r$$

with $P \in \mathbb{R}^{n \times n}$, $q \in \mathbb{R}^n$, and $r \in \mathbb{R}$. Since $\nabla^2 f(x) = P$ for all x, f is convex if and only if $P \succeq 0$ (and concave if and only if $P \preceq 0$).

Exercise: Consider the mean-variance portfolio example in the previous section. For what values of θ is the objective function convex? concave?

3.3 Further examples

We saw that all linear functions are convex as well as concave, and have described the convex and concave quadratic functions. In this section we give a few more examples of convex and concave functions. We start with some functions on $\mathbb{R}$, with one-dimensional variable x.

- Exponential e^{ax} is convex on $\mathbb{R}$, for any $a \in \mathbb{R}$.
- Powers x^a is convex on $\mathbb{R}_+$ when $a \geq 1$ or $a \leq 0$, and concave for $0 \leq a \leq 1$.
- Powers of absolute value. x^p, for $p \geq 1$, is convex on $\mathbb{R}_+$. x^p, for $0 \leq p \leq 1$, is concave on $\mathbb{R}_+$.
- Logarithm. $\log(x)$ is concave on $\mathbb{R}_+$.

(Verify using either the definition of covnex/concave functions or the condition on second derivative fo twice differentiable functions)

We now give a few interesting examples of functions on $\mathbb{R}^n$.

- Norms. Every norm on $\mathbb{R}^n$ is convex.
- Max function. $f(x) = \max\{x_1, \ldots, x_n\}$ is convex on $\mathbb{R}^n$.
- Least squares objective: $||Ax - b||^2$ is convex is convex on $\mathbb{R}^n$.
- Quadratic-over-linear function. The function $f(x, y) = x^2/y$, with $dom(f) = R \times \mathbb{R}^+ = (x, y) \in \mathbb{R}^2 | y > 0$ is convex.
- log-sum-exp. $f(x) = \log(e^{x_1} + \cdots e^{x_n})$ is convex on $\mathbb{R}^n$.

Some operations that preserve convexity:

- Pointwise maximum
- Non-negative weighted sum
- Affine composition ($g(x) = f(Ax + b)$)

Exercise: Consider the loss functions in the linear regression application discussed in the previous section. Which of those are convex? concave?

4 Convex optimization: Optimality conditions

Consider the problem of minimizing a convex function (or equivalently maximizing a concave function, since concave functions are simply negative of a convex function).

Differentiable objective

Theorem 4.1. *Suppose that the objective f in a convex program is differentiable, Let C denote the feasible convex set. Then an x in C is optimal if and only if for all $y \in C$,*

$$\nabla f(x)^T (y - x) \geq 0$$

If the condition is satisfied for all y, then clearly (by convexity of f)

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \geq f(x),$$

and therefore, x must be optimal. For the other direction, suppose that a $y \in C$ exists that does not satisfy this condition for x, i.e., $\nabla f(x)^T (y - x) < 0$ for this y. Now, by convexity of set C, all the points on the line segment joining x and y are in C. Consider a point $tx + (1 - t)y$ close to x ($t \to 1$) on this line segment, then

$$f(tx + (1 - t)y) - f(x) \simeq \nabla f(x)^T ((1 - t)y - (1 - t)x) < 0.$$

Therefore, x cannot be optimal.

Unconstrained problems. For an unconstrained problem, the optimality condition reduces to

$$\nabla f(x) = 0 \tag{11}$$

To see intuitively how it follows from Theorem 4.1, suppose that x is optimal, which means for all feasible y we have $\nabla f(x)^T(y-x) \geq 0$. Since f is differentiable, its domain is (by definition) open, so all y sufficiently close to x are feasible. Let us take $y = x - t\nabla f(x)$. For t small and positive, y is feasible, and so for optimality of x it must hold that $\nabla f(x)^T(y-x) = -t\nabla f(x)^T\nabla f(x) \geq 0$, from which we conclude $\nabla f(x) = 0$.

Example: Unconstrained quadratic optimization. Consider the problem of minimizing the quadratic function $f(x) = (1/2)x^TPx + qTx + r$, where P is an $n \times n$ matrix, and $P \succeq 0$. The necessary and sufficient condition for x to be a minimizer of f is

$$\nabla f(x) = Px + q = 0.$$

5 Unconstrained convex optimization: Gradient Descent algorithm

Here, we consider the problem of minimizing a differentiable convex function $f(x)$ over $\mathbb{R}^n$.

Consider the following simple scheme for optimization: Start at some point x_0, and in every step move from the current point x to a new point $x + \Delta x$ such that $f(x + \Delta x) < f(x)$. Such directions (Δx) are called 'descent directions'. Stop when you reach the optimal point ($\nabla f(x) = 0$). At this point, no descent direction exists.

For a one-dimensional function, there are only two directions: forward or backward. And, for a convex function (trough-shaped), at any sub-optimal point, one of these two directions is a descent direction. In fact, at the points where the gradient is negative (left of the minima), stepping in the forward direction, or taking a positive step, decreases the function value ($f(x + \eta) < f(x)$ for small η). At the points where gradient is positive (left of the minima), stepping in the backward direction, or taking a negative step, decreases the function value ($f(x-\eta) < f(x)$ for small η). Therefore, the sign of the "step" should be opposite the sign of the gradient.

For a multivariate function, there can be many descent directions. A natural choice is again the negative gradient $\Delta x = -t\nabla f(x)$. This is always a descent direction for small enough step size $t > 0$: to see this intuitively, observe that for very small t, using Taylor expansion,

$$f(x + \Delta x) \simeq f(x) + \nabla f(x)^T\Delta x = f(x) - t\nabla f(x)^T\nabla f(x) < f(x),$$

for any suboptimal point x. In one dimension:

$$f(x + \Delta x) \simeq f(x) + f'(x)\Delta x = f(x) - tf'(x)^2 < f(x)$$

for any suboptimal point x.

This choice of descent direction results in the popular 'Gradient Descent method'.

Gradient descent method. Given a starting point x in domain of function f.
repeat the following steps:

1. Choose a step size t.
2. Set descent direction $\Delta x := -\nabla f(x)$.
3. Update. $x := x + t\Delta x$.

until a stopping criterion $\nabla f(x) \leq \epsilon$ is satisfied.

The choice of step size t is important here. One would like a larger step size in order to be able to move through the space faster. However, descent is guaranteed only for a small enough t; with a larger step size t, one could even end up increasing the function value, or not decrease it 'enough'. (draw a one dimensional convex function, and try using different step sizes in the descent direction from a sub-optimal point x. Does the function value always decrease? Does larger step size always result in more decrease?)

Given a descent direction Δx, one could do a comprehensive search for the 'best' step size, also known as 'exact line search':

$$t = \arg\min_{s \geq 0} f(x + s\Delta x)$$

An exact line search is used when the cost of the minimization problem with one variable, is low compared to the cost of computing the search direction itself.

In practice, inexact methods are used to find a step size that reduces f 'enough'. A popular heuristic for choosing step size is Backtracking line search. This heuristic starts with step size 1, and decreases it by a factor of β until it finds a step size t which provides enough reduction: $f(x) - f(x + t\Delta x)$ is atleast $\alpha t |\nabla f(x)^T \Delta x| = |\alpha t \nabla f(x)^T \nabla f(x)|$ for some constant $\alpha < 1/2$. Using Taylor expansion, there always exist a small enough t which can achieve this reduction.

Backtracking line search. Given a descent direction Δx (e.g., $\Delta x = -\nabla f(x)$) for f at x, $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$. Set $t := 1$.
while $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^T \Delta x$, update $t := \beta t$.

5.1 An example in $\mathbb{R}^2$

Consider unconstrained minimization of convex function

$$f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2).$$

with $\gamma > 0$. Clearly the optimal solution is $x_1 = 0, x_2 = 0$. Let's examine the execution of gradient descent algorithm for this problem with exact and backtracking line search, starting from point $(\gamma, 1)$. Here,

$$\nabla f(x) = \begin{bmatrix} x_1 \\ \gamma x_2 \end{bmatrix}$$

Therefore, given a point (x_1, x_2), the next point in gradient descent algorithm will be

$$x'(t) = (x_1 - tx_1, x_2 - t\gamma x_2) = ((1-t)x_1, (1-\gamma t)x_2)$$

where t is the step size. Value at next iterate:

$$f(x'(t)) = \frac{(1-t)^2}{2} x_1^2 + \frac{(1-\gamma t)^2 \gamma}{2} x_2^2.$$

Exact line search Set t as:

$$\arg\min_{t \geq 0} f(x'(t))$$

This is a one dimensional minimization problem in t, solution is t such that

$$(1-t)x_1^2 + (1-t\gamma)\gamma^2 x_2^2 = 0$$

$$t = \frac{x_1^2 + \gamma^2 x_2^2}{x_1^2 + \gamma^3 x_2^2}$$

For first step $x_1 = \gamma, x_2 = 1$, so,

$$t = \frac{2\gamma^2}{\gamma^2(1+\gamma)} = \frac{2}{1+\gamma}, 1 - t = \frac{\gamma - 1}{\gamma + 1}, 1 - \gamma t = -\frac{\gamma - 1}{\gamma + 1}$$

$$x^{(1)} = ((1-t)x_1, (1-\gamma t)x_2) = \frac{\gamma(\gamma - 1)}{\gamma + 1}, -\frac{\gamma - 1}{\gamma + 1}$$

In fact,

$$x^{(k)} = \gamma \left(\frac{\gamma-1}{\gamma+1}\right)^k, \left(-\frac{\gamma-1}{\gamma+1}\right)^k$$

(Verify)

Therefore, (letting $z := \left(\frac{\gamma-1}{\gamma+1}\right)$)

$$f(x^{(k)}) = \frac{1}{2}(\gamma^2 z^{2k} + \gamma(-z)^{2k}) = \frac{1}{2}\gamma(\gamma+1)z^{2k} = z^{2k} f(x^{(0)})$$

where $x^0 = (\gamma, 1)$ is the starting point, with $f(x^0) = \frac{1}{2}(\gamma^2 + \gamma)$.

From above we can observe that if γ is close to 1, so that z is close to 0, then the convergence to 0 is fast. The convergence is slow if $\gamma >> 1$ or $\gamma << 1$.

5.2 Backtracking vs. Exact line search

The convergence for backtracking line search is slower than exact line search (but computation of step size is faster with backtracking line search).

The figure below compares the execution of gradient descent method for a convex minimization problem with $f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$.

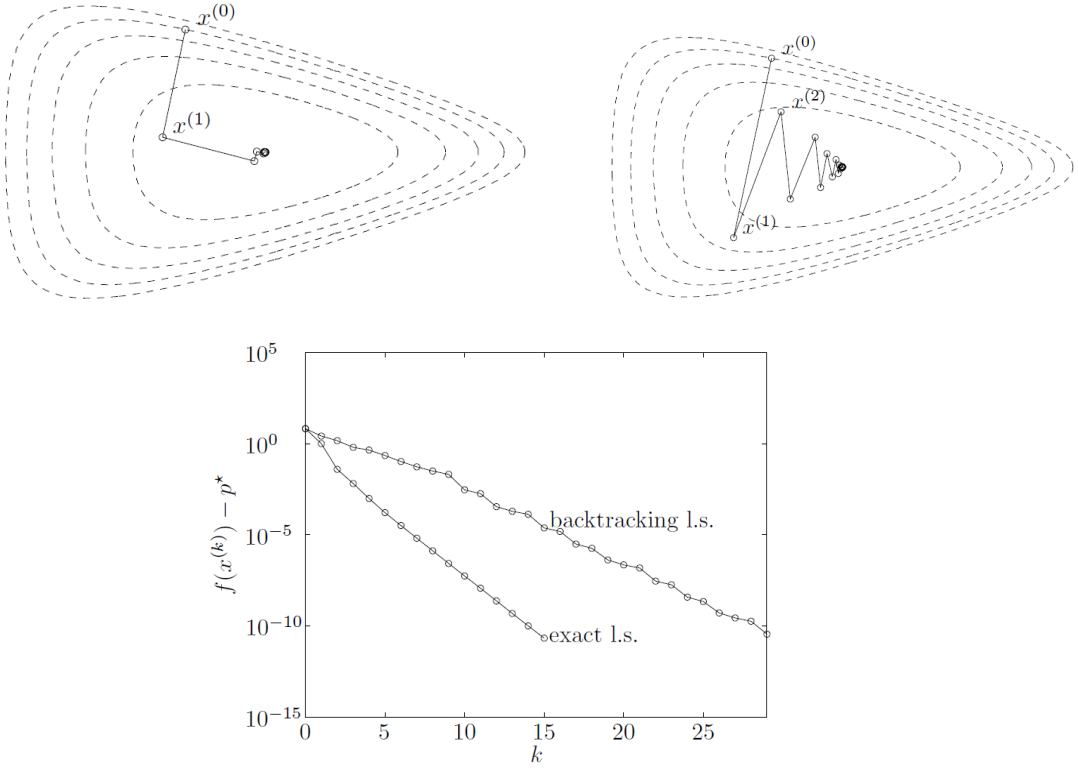


Figure 6: (a) Exact Line Search vs. (b) Backtracking line search for function $f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$

6 Linearly constrained convex optimization

The previous sections (and the example on training neural networks) all concerned unconstrained optimization. However in section 1 we saw cases of optimization problems with a convex objective but linear constraints, e.g. problem (2) related to mean-variance portfolio optimization (and also see the writeup on portfolio optimization). In this section we discuss techniques that extend the approach used to handle unconstrained problems to linearly constrained problems. Such problems can be summarized as

$$\text{Minimize } F(x) \tag{12a}$$
$$\text{subject to:} \tag{12b}$$
$$Ax \geq b \tag{12c}$$

Here, the function $F(x)$ is assumed to be convex, and $Ax \geq b$ is a generic way to write linear constraints. In terms of example (2),

$$F(x) = -f(x) = -20x_1 - 16x_2 + \theta[2x_1^2 + x_2^2 + (x_1 + x_2)^2]$$

[For a minimization problem the function should be convex if we want to use convex optimization techniques. Exercise: verify that that F(x) given above is convex]. And $Ax \geq b$ stands for constraints (2c), (2d).

In this section we will describe a generalization of the gradient descent methods described above, which applies to general problems of the form (12). We will then outline an example.

The idea in this the algorithm is that we will iterate and at each iteration find a better solution. We begin by finding a feasible solution x^1 (i.e. $Ax^0 \geq b$) which can be done for example by applying the Phase I method of linear programming. At the k^{th} iteration ($k = 1, 2, \ldots$) we have a feasible solution x^k (i.e. $Ax^k \geq b$) and we compute a "perturbation" to x^k that improves on the F value. In detail, the algorithm is as follows (explanations, after) where we assume that the vectors x are n-dimensional:

Step 0. Compute x^1 such that $Ax^1 \geq b$.

Step k. [Here $k = 1, 2, \ldots$]. Define $g^k = \nabla F(x^k)$. Solve the linear program with variables v^k:

$$\min \quad (g^k)^T v^k \tag{13}$$
$$\text{subject to:} \quad Av^k \geq b - Ax^k \tag{14}$$
$$-1 \leq v_j^k \leq 1, \quad 1 \leq j \leq n \tag{15}$$

Suppose v^* is the optimal solution[2].
(a) If $g^{kT} v^* = 0$ **STOP**: x^k is **optimal** for problem (12).
(b) Otherwise $g^{kT} v^* < 0$.
(c) Perform a line-search to compute a step-size $0 < \sigma_k \leq 1$. The next iterate, x^{k+1} will be defined by

$$x^{k+1} = x^k + \sigma_k v^*$$

.

Now let us analyze this algorithm.

- Notice that (14) can be rewritten as
$$A(x^k + v^k) \geq b$$
which means that $x^k + v^k$ remains feasible. Thus, setting $v^k = 0$ yields a feasible solution to the LP (13), since x^k is feasible. So the value of the LP (13) is either 0, or is negative. The reason for constraint (15) is twofold: first, it guarantees that the LP (13) is bounded and second, it guarantees that the vector $x^k + v^k$ is "near" x_k.
- If x^k is **not** an optimal solution to the optimization problem (12) then one can easily argue (using convexity) that the value of the LP **must** be negative. So **(a)-(b)** in the algorithm are correct.

[2]Note, v^* depends on the iteration index k, but we are skipping that for simplicity

- Assume that we have condition **(b)**. Notice that $v^*/\|v^*\|$ is a unit-norm vector in the direction of v^*. As we discussed in Section 3.1, the inner product

$$(g^k)^T \frac{v^*}{\|v^*\|}$$

(which by (b) is negative) is the **directional derivative** at x^k in the direction of v^*. **Thus, condition (b) indicates that as we move away from x^k in the direction of v^*, the function F is decreasing**.

- Thus, there remains to compute **how far** to move away from x^k in the direction of v^*. Step (c) takes care of that issue.

Example: Consider the problem

$$\begin{aligned} \text{Minimize } & -6x_1 + e^{x_1/3+x_2} \\ \text{subject to: } & x_1 \geq 5,\ x_2 \geq 1 \end{aligned}$$

Note that

$$\nabla F(x) \ = \ \left(-6 + \frac{e^{x_1/3+x_2}}{3}\,,\, e^{x_1/3+x_2}\right)^T$$

[Verify that the objective is convex]
Step 0. Set $x^1 = (5,1)^T$. (For example). So we have that $F(x^1) \approx -15.6081$.
Step 1. We have that $g^1 = \nabla F(x^1) = (-1.20269, 14.39192)^T$. So the LP to solve is:

$$\begin{aligned} \text{Minimize } & -1.20269\, v_1 + 14.39192\, v_2 \\ \text{subject to: } & v_1 \geq 0,\ v_2 \geq 0 \\ & -1 \leq v_1 \leq 1,\ -1 \leq v_2 \leq 1 \end{aligned}$$

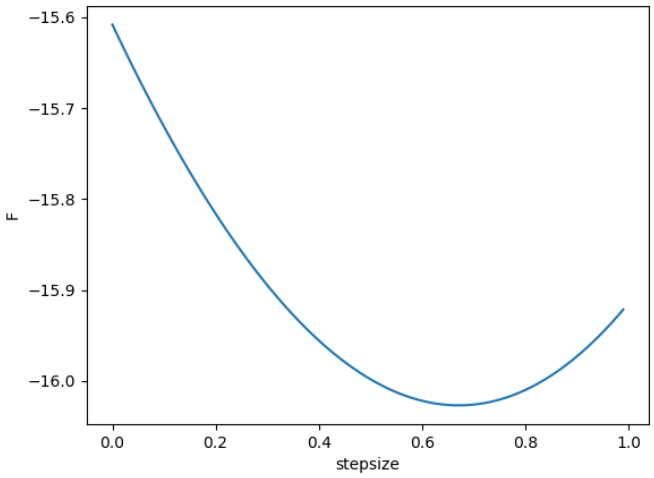
The optimal solution to this LP is: $v_1^* = 1.0, v_2^* = 0$, and the value of the LP is -1.20269. Hence we have case (b), i.e. the direction given by v^* is a descent direction, namely that if we set

$$x^2 \ \leftarrow \ x^1 + \sigma^1 v^*,$$

for $0 < \sigma^1$ *small enough* we will have $\boldsymbol{F(x^2) < F(x^1)}$. Step (c) will compute an appropriate step-size.

Exercise: If we set $0 \leq \sigma^1 \leq 1$, *why is x^2 feasible in the general case?*

Here is a plot of the values $F(x^1 + \sigma v^*)$:



References

[1] 'Applied Mathematical Programming' by Bradley, Hax, and Magnanti (Addison-Wesley, 1977).

[2] 'Convex Optimization' by Stephen Boyd and Lieven Vandenberghe, Cambridge University Press.

Wed.Apr..4.144128.2018@blacknwhite