Unit 4

1 Convexity

Let us start with the definition of convex and strictly convex function.

Definition 1.1 *(Convex function)*
A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y),$$

for any $x, y \in \mathbb{R}^n$ and any $\lambda \in [0, 1]$.

Definition 1.2 *(Strictly convex function)*
A function $f : \mathbb{R}^n \to \mathbb{R}$ is strictly convex if

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y),$$

for any $x, y \in \mathbb{R}^n$, with $x \neq y$, and any $\lambda \in (0, 1)$.

The following is an algebraic characterization of convex and strictly convex function.

Proposition 1.3 *A function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if and only if*

$$v^T \nabla^2 f(x) v \geq 0, \quad \forall\, x, v \in \mathbb{R}^n,$$

that is the hessian matrix $\nabla^2 f(x)$ is positive semidefinite for all $x \in \mathbb{R}^n$.
The function $f : \mathbb{R}^n \to \mathbb{R}$ is strictly convex if

$$v^T \nabla^2 f(x) v > 0, \quad \forall\, x, v \in \mathbb{R}^n,\ v \neq 0,$$

that is the hessian matrix $\nabla^2 f(x)$ is positive definite for all $x \in \mathbb{R}^n$.

The following is a geometric definition of convex set.

Definition 1.4 *(Convex set)*
A set $\mathcal{S} \subseteq \mathbb{R}^n$ is convex if $[y, z] \subseteq \mathcal{S}$ for any $y, z \in \mathcal{S}$.

The following proposition gives sufficient conditions to guarantee that a set is convex.

Proposition 1.5 *Let $\mathcal{S} \triangleq \{x \in \mathbb{R}^n\ :\ g_j(x) \leq b_j,\ j = 1, \dots, m\}$. If all the functions g_j are convex, then $\mathcal{S}$ is convex.*

Proof. The set $\mathcal{S}$ is the intersection of m sets:

$$\mathcal{S} = \cap_{j=1}^m \mathcal{S}_j, \quad \mathcal{S}_j \triangleq \{x \in \mathbb{R}^n \,:\, g_j(x) \le b_j\}, \;\; j = 1, \ldots, m.$$

Let us consider a generic j, and let $y, z \in \mathcal{S}_j$. For any $\lambda \in [0,1]$, by the convexity of g_j, we get

$$b_j = \lambda b_j + (1-\lambda) b_j \ge \lambda g_j(y) + (1-\lambda) g_j(z) \ge g_j(\lambda y + (1-\lambda) z).$$

Therefore $[y, z] \subseteq \mathcal{S}_j$, that is, the set $\mathcal{S}_j$ is convex.

Let $y, z \in \mathcal{S}$. Therefore $y, z \in \mathcal{S}_j$ for any $j = 1, \ldots, m$. Therefore, by the convexity of any $\mathcal{S}_j$, we get $[y, z] \subseteq \mathcal{S}_j$ for any $j = 1, \ldots, m$. Finally $[y, z] \subseteq \mathcal{S}$, that is, $\mathcal{S}$ is convex. $\square$

1.1 Quadratic functions and convexity

Let $f(x) \triangleq \frac{1}{2} x^T Q x + s^T x + t$ be a quadratic function:

- its gradient vector is: $\nabla f(x) = Qx + s$,
- its hessian matrix is: $\nabla^2 f(x) = Q$,
- it is convex if and only if Q is positive semidefinite ($Q \succeq 0$),
- it is strictly convex if **and only if** Q is positive definite ($Q \succ 0$).

Let

$$Q = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & f \end{pmatrix}.$$

The matrix Q is positive semidefinite if and only if all its principal minors are non negative, that is, if and only if all the following conditions are satisfied:

- $a \ge 0$, $d \ge 0$, $f \ge 0$
- $\det \begin{pmatrix} a & b \\ b & d \end{pmatrix} = ad - b^2 \ge 0$, $\det \begin{pmatrix} a & c \\ c & f \end{pmatrix} = af - c^2 \ge 0$, $\det \begin{pmatrix} d & e \\ e & f \end{pmatrix} = df - e^2 \ge 0$
- $\det Q = a(df - e^2) - b(bf - ce) + c(be - cd) \ge 0$.

The matrix Q is positive definite if and only if all its principal minors of north-west are positive, that is, if and only if all the following conditions are satisfied:

- $a > 0$
- $\det \begin{pmatrix} a & b \\ b & d \end{pmatrix} = ad - b^2 > 0$
- $\det Q = a(df - e^2) - b(bf - ce) + c(be - cd) > 0$.

2 NonLinear Programming

A continuous problem is in general a NonLinear Program (NLP):

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & g_j(x) \le b_j, \quad j = 1, \dots, m \\ & x \in \mathbb{R}^n. \end{aligned}$$

We recall the notation: $\mathcal{S} \triangleq \{x \in \mathbb{R}^n \,:\, g_j(x) \le b_j,\, j = 1, \dots, m\}$.

It is useful to define a relaxed version of optimality:

- $\overline{x}$ is said to be a **local optimal point** if $\overline{x} \in \mathcal{S}$ (it is feasible) and an $\varepsilon > 0$ exists such that
$$f(\overline{x}) \le f(x), \quad \forall\, x \in \mathcal{S} \cap \{z \in \mathbb{R}^n \,:\, \|z - \overline{x}\| < \varepsilon\}.$$

One and only one of the following statements is true:

(i) the NLP is infeasible,

(ii) the NLP is unbounded,

(iii) the NLP admits optimal points,

(iv) the NLP is bounded, but does not admit optimal points.

Remarks:

- Case (iv) can occur only when some functions are nonlinear. In fact, otherwise the fundamental theorem of linear programming rules out case (iv).
- Local optimal points can occur in cases (ii), (iii), and (iv).

[Give practical simple examples in which all these cases occurs.]

2.1 Convex problems

Definition 2.1 *The NLP is a* **convex problem** *if f is a convex function and $\mathcal{S}$ is a convex set.*

Definition 2.2 *The NLP is a* **strictly convex problem** *if f is a strictly convex function and $\mathcal{S}$ is a convex set.*

Theorem 2.3 *In a convex problem all local optimal points are (global) optimal points.*

Theorem 2.4 *A strictly convex problem either does not admit any local optimal point, or it admits a unique local optimal point, and this is also a (global) optimal point.*

Remarks:

- LPs are convex, but not strictly convex, problems. Theorem 2.3 explains why the point given by the simplex algorithm is actually an optimal point of the LP.
- ILPs are not convex problems.

2.2 Convex quadratic-linear problems

Consider the quadratic-linear problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \frac{1}{2}x^T Q x + s^T x + t \\ \text{subject to} \quad & \sum_{i=1}^{n} a_{ij} x_i \le b_j, \quad j = 1, \dots, m \\ & x \in \mathbb{R}^n, \end{aligned}$$

where $Q \succeq 0$, that is the problem is convex.

A point $\overline{x}$ is an optimal point of the problem if and only if a vector $\lambda \in \mathbb{R}^m$ exists such that

$$Q\overline{x} + s + \sum_{j=1}^{m} \lambda_j a_{*j} = 0$$

$$\sum_{i=1}^{n} a_{ij} x_i \le b_j, \quad \lambda_j \ge 0, \quad \lambda_j \left(\sum_{i=1}^{n} a_{ij} x_i - b_j \right) = 0, \quad j = 1, \dots, m.$$

Remarks:

- These conditions are called Karush-Kuhn-Tucker (KKT) conditions.
- These conditions include some nonlinear equalities.

The KKT conditions can be equivalently rewritten by using some binary variables $\delta \in \{0,1\}^m$:

$$Q\overline{x} + s + \sum_{j=1}^{m} \lambda_j a_{*j} = 0$$

$$\sum_{i=1}^{n} a_{ij} x_i \le b_j, \quad \lambda_j \ge 0, \quad j = 1, \dots, m$$

$$\sum_{i=1}^{n} a_{ij} x_i \ge b_j - M\delta_j, \quad \lambda_j \le M(1 - \delta_j), \quad \delta_j \in \{0,1\}, \quad j = 1, \dots, m,$$

where $M > 0$ is a huge number. Remarks:

- Any condition $\lambda_j \left(\sum_{i=1}^n a_{ij}x_i - b_j\right) = 0$ is viewed as a disjunctive constraint ($\lambda_j = 0$ or $\sum_{i=1}^n a_{ij}x_i - b_j = 0$), and then modified by using the binary variables δ (see unit 3).

- These conditions include only linear constraints. Therefore the problem of computing a pair $(\overline{x}, \lambda)$ satisfying these conditions can be viewed as a MILP with a constant objective function and a feasible set made up by these conditions.

3 Regression analysis

Regression is an approach to modeling the relationships between an input $z \in \mathbb{R}^n$ and an output $y \in \mathbb{R}^p$. These relationships are modeled using a predictor function $\theta_\omega : \mathbb{R}^n \to \mathbb{R}^p$ whose unknown model parameters $\omega \in \mathbb{R}^s$ are estimated from some observed input/output data (z^i, y^i), $i = 1, \dots, d$. Regression has many practical uses. Most applications fall into one of the following two broad categories:

- If the goal is prediction, or forecasting, regression can be used to fit a predictive model to an observed data set of values of input and output. After developing such a model, if additional values of the input are collected without some accompanying output values, the fitted model can be used to make a prediction of the output.

- If the goal is to explain variations in the output that can be attributed to variation in the input, regression analysis can be applied to quantify the strength of the relationship between the output and the input, and in particular to determine whether some specific input z_k may have no relationship with the output at all, or to identify which subsets of input variables may contain redundant information about the output.

Regression models are often fitted using the least squares approach. This consists in minimizing the following error function

$$\underset{\omega}{\text{minimize}} \quad \frac{1}{2} \sum_{i=1}^{d} \sum_{j=1}^{p} \left(\theta_\omega(z^i)_j - y^i_j\right)^2 .$$

3.1 Linear regression

We assume that the predictor function is linear and there is a unique output ($p = 1$). In this case we have n parameters ω_k to estimate:

$$\theta_\omega(x) \triangleq \sum_{k=1}^{n} \omega_k z_k .$$

The error function to minimize turns out to be the following:

$$\underset{\omega}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{d}\left(\sum_{k=1}^{n}\omega_k z_k^i - y^i\right)^2 = \frac{1}{2}\omega^T Q\omega + s^T\omega + t,$$

where

$$Q = \begin{pmatrix} z^1 & \cdots & z^d \end{pmatrix}\begin{pmatrix} z^1 & \cdots & z^d \end{pmatrix}^T, \qquad s = -\begin{pmatrix} y^1 & \cdots & y^d \end{pmatrix}\begin{pmatrix} z^1 & \cdots & z^d \end{pmatrix}^T,$$

$$t = \frac{1}{2}\begin{pmatrix} y^1 & \cdots & y^d \end{pmatrix}\begin{pmatrix} y^1 & \cdots & y^d \end{pmatrix}^T.$$

Following the method described in section 2.2, the parameters ω can be estimated by solving the following KKT conditions:

$$Q\omega + s = 0.$$

This model written in GNU MathProg is reported in the file `linear_regression1.mod`:

```
param n;
param d;

param z{1..d,1..n};
param y{1..d};

var omega{1..n};

s.t. grad_obj {k in 1..n}:
  sum{kk in 1..n} sum{i in 1..d} z[i,k]*z[i,kk]*omega[kk] -
  sum{i in 1..d} z[i,k]*y[i] = 0;


display z;
display y;

solve;

display omega;
display sum{i in 1..d} ( sum{k in 1..n} z[i,k]*omega[k] - y[i] )^2;

end;
```

Let us consider the data

y	19	3	4	13	10	7	6	11	6	2
	9	2	2	10	6	4	3	4	2	1
z	9	2	3	5	5	3	1	7	5	1
	1	0	1	1	0	1	1	0	1	0

This data set is reported in the file `linear_regression1.dat`:

```
param n := 3;
param d := 10;

param z: 1 2 3 :=
1 9 9 1
2 2 2 0
3 2 3 1
4 10 5 1
5 6 5 0
6 4 3 1
7 3 1 1
8 4 7 0
9 2 5 1
10 1 1 0;

param: y :=
1 19
2 3
3 4
4 13
5 10
6 7
7 6
8 11
9 6
10 2;

end;
```

The output file:

```
Problem:    linear_regression1
Rows:       3
Columns:    3
Non-zeros:  9
Status:     OPTIMAL
Objective:  0 (MINimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 grad_obj[1]  NS           479           479             =         < eps
     2 grad_obj[2]  NS           440           440             =         < eps
     3 grad_obj[3]  NS            55            55             =         < eps
```

```
   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 omega[1]     B       0.905438
     2 omega[2]     B       0.994127
     3 omega[3]     B       0.331596

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err = 5.68e-14 on row 1
        max.rel.err = 1.28e-16 on row 3
        High quality

KKT.PB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

KKT.DE: max.abs.err = 0.00e+00 on column 0
        max.rel.err = 0.00e+00 on column 0
        High quality

KKT.DB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

End of output
```

Exercise: Modify the model by adding a constraint imposing that $\omega_2 + \omega_3 \leq 1$ and solve it with GLPK.
[Produce the files `linear_regression2.mod` and `linear_regression2.dat`.]

3.2 Nonlinear regression

Assume that the predictor function is nonlinear and there is a unique output ($p = 1$). Let us consider, for example, the case in which we have $2n$ parameters ω_k to estimate:

$$\theta_\omega(x) \triangleq \sum_{k=1}^{n} \omega_k z_k + \sum_{k=1}^{n} \omega_{k+n}(z_k)^2.$$

The error function to minimize turns out to be the following:

$$\underset{\omega}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{d}\left(\sum_{k=1}^{n}\omega_k z_k^i + \sum_{k=1}^{n}\omega_{k+n}(z_k^i)^2 - y^i\right)^2 = \frac{1}{2}\omega^T Q\omega + s^T\omega + t,$$

where

$$Q = NN^T, \qquad s = -\begin{pmatrix} y^1 & \cdots & y^d \end{pmatrix} N^T, \qquad t = \frac{1}{2}\begin{pmatrix} y^1 & \cdots & y^d \end{pmatrix}\begin{pmatrix} y^1 & \cdots & y^d \end{pmatrix}^T,$$

$$N = \begin{pmatrix} z_1^1 & \cdots & z_1^d \\ \vdots & & \vdots \\ z_n^1 & \cdots & z_n^d \\ (z_1^1)^2 & \cdots & (z_1^d)^2 \\ \vdots & & \vdots \\ (z_n^1)^2 & \cdots & (z_n^d)^2 \end{pmatrix}$$

This problem can be solved in the same way as before.

```
param n;
param d;

param z{1..d,1..n};
param y{1..d};

var omega{1..2*n};

s.t. grad_obj1 {k in 1..n}:
  sum{kk in 1..n} sum{i in 1..d} z[i,k]*z[i,kk]*omega[kk] +
  sum{kk in 1..n} sum{i in 1..d} z[i,k]*(z[i,kk]^2)*omega[n+kk] -
  sum{i in 1..d} z[i,k]*y[i] = 0;

s.t. grad_obj2 {k in 1..n}:
  sum{kk in 1..n} sum{i in 1..d} (z[i,k]^2)*z[i,kk]*omega[kk] +
  sum{kk in 1..n} sum{i in 1..d} (z[i,k]^2)*(z[i,kk]^2)*omega[n+kk] -
  sum{i in 1..d} (z[i,k]^2)*y[i] = 0;

display z;
display y;

solve;

display omega;
display sum{i in 1..d} ( sum{k in 1..n} z[i,k]*omega[k] +
        sum{k in 1..n} (z[i,k]^2)*omega[n+k] - y[i] )^2;

end;
```