Optimization

Exercice 1 *Calculus.*

1. (2 points) Find the first and second derivatives of the composed exponential function

$$f(x) = e^{-x^2}$$

2. (2 points) Find the first derivative of the variable base and variable exponent function

$$f(x) = x^x$$

3. (2 points) Find the gradient of the multi-dimensional input function

$$f(x, y, z) = 2^{xy} + z\cos(x)$$

Exercice 2 *Convex Functions.*

1. (5 points) Derive the 2×2 symmetric matrix whose eigenvalues are 7 and 1, such that $(1, 1)$ is an eigenvector with eigenvalue 7. *Hint: The second eigenvector is orthogonal to the first one.*

2. (5 points) Is the function $f(x_1, x_2) = x_1^4 + 2x_1^2 + 3x_1x_2 + 2x_2^2 - 7x_1 - 12x_2 - 18$ convex? Justify your answer. *Hint: What is the condition on the Hessian for f to be convex?*

3. (5 points) Consider a cost function $J(\mathbf{w})$ over a weight vector $\mathbf{w}$, and suppose that at every point $\mathbf{w} \in \mathbb{R}^p$, the Hessian matrix $\nabla^2 J$ is positive definite. Is it always true that $J(\mathbf{w})$ has exactly one unique local minimum $\mathbf{w}^\star \in \mathbb{R}^p$? Why or why not?

Exercice 3 *Saddle point.* Show that the function $f(\mathbf{x}) = 8x_1 + 12x_2 + x_1^2 - 2x_2^2$ has only one stationary point, and that it is neither a maximum or minimum, but a saddle point. Sketch the contour lines of f.

Plotting methods can be found on the Notebooks on discord

Exercice 4 *Rosenbrock function.*

1. Compute the gradient $\nabla f(\mathbf{x})$ and Hessian $\nabla^2 f(\mathbf{x})$ of the Rosenbrock function

$$f(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

2. Show that $\mathbf{x}^\star = (1, 1)^T$ is the only local minimizer of this function, and that the Hessian matrix at that point is positive definite.

3. You will find on discord a notebook named `TD 2-Multivariate-Optimization.ipynb`

   1. On the notebook, complete the cells needed (i.e. '`df(x)`') and check the code in the '`gradient_descent`' function. Compare with the pseudo-code given in the slides.
   2. Check the '`gradient_descent`' by using different starting point $\mathbf{x}_0$ and compare the number of iterations : What do you observe when you start far from $\mathbf{x}^\star$?
   3. Implement the Newton algorithm and compare with the previous results.

Exercice 5 *Camel-Hump function.* Same questions with the Camel-Hump function

$$f(x,y) = 2x^2 - 1.05x^4 + \frac{x^6}{6} + xy + y^2$$

Modify the starting points of the algorithms and compare the results. Conclusion ?

Zero Inflated Poisson (ZIP) distribution

A random variable X following the usual Poisson distribution with parameter λ, $\mathcal{P}(\lambda)$, with the probability mass function

$$\mathbb{P}(X = k) = \exp(-\lambda)\frac{\lambda^k}{k!}, \quad k = 0, 1, \ldots \tag{1}$$

is widely used to model many naturally occurring events where X represents the "number of events per unit of time or space". Note that X takes only nonnegative integer values. However, the $\mathcal{P}(\lambda)$ distribution may not be useful (or it gives a bad fit) when X takes the value 0 with a high probability.

In such a case a modified version of a regular $\mathcal{P}(\lambda)$ distribution known as the zero-inflated Poisson (ZIP) distribution becomes useful. The ZIP distribution with parameters π and λ, denoted by ZIP(π, λ), has the following probability mass function:

$$\mathbb{P}(X = k) = \begin{cases} \pi + (1-\pi)\exp(-\lambda) & \text{if } k = 0 \\ (1-\pi)\exp(-\lambda)\frac{\lambda^k}{k!} & \text{if } k = 1, 2, \ldots \end{cases} \tag{2}$$

This model represents a **mixture** between a discrete distribution such that $\mathbb{P}(X = 0) = 1$ with probability π and a classical Poisson distribution with parameter λ with probability $1 - \pi$.

1. First results

   1. What is the ML estimator of λ in a Poisson model given a sample $(x_i)_{i=1}^n$? [1]
   2. Show that the likelihood of the parameters of a ZIP distribution for a sample $(x_i)_{1\le i\le n}$ assumed to come from such a distribution is

   $$L_n(\pi, \lambda; (x_i)_{1\le i\le n}) = \prod_{i=1}^{n} \left(\pi + (1-\pi)e^{-\lambda}\right)^{\mathbf{1}_{x_i=0}} \left((1-\pi)\frac{\lambda^{x_i}}{x_i!}e^{-\lambda}\right)^{\mathbf{1}_{x_i\neq 0}}$$

   3. Denoting n_0, the number of sample equal to zero, and $s_n = \sum_{i=1}^n x_i$ the sum of the sample, show that the expression of the log-likelihood $l(\pi, \lambda; (x_i)_{1\le i\le n})$ is

   $$l_n(\pi, \lambda; (x_i)_{1\le i\le n}) = n_0 \ln(\pi + (1-\pi)e^{-\lambda}) + (n-n_0)\ln(1-\pi) - (n-n_0)\lambda + s_n \ln(\lambda) - \ln \prod_{i=}^{n} x_i!$$

   4. Implement the computation of the log-likelihood $l(\pi, \lambda; (x_i)_{1\le i\le n})$

   ```
   # loglikelihood function
   def ln(pi, lam, X):
       n  = X.shape[0]
       n0 = np.sum(X==0)
       sn = np.sum(X)
       return n0*np.log(pi+(1-pi)*np.exp(-lam))\\
               + (n-n0)*np.log(1-pi)-(n-n0)*lam \\
               + sn*np.log(lam)
   ```

[1] You should find $\hat{\lambda} = \frac{1}{n}\sum_{i=1}^n x_i$

2. You will find on discord a notebook with a function `simzip(pi, l, size)` and a function `probzip` that simulates independent realizations of a ZIP distribution with parameters π and λ and compute the probabilities of occurrence of the events.

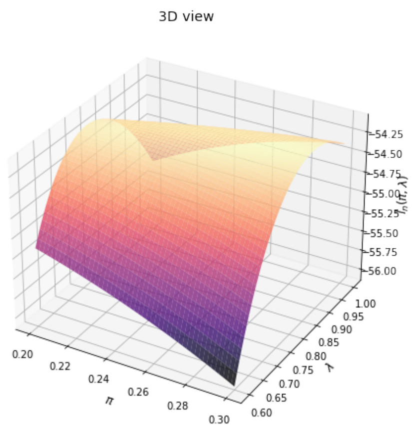
   Verify the code by comparing the frequencies obtained by simulation with the theoretical values.

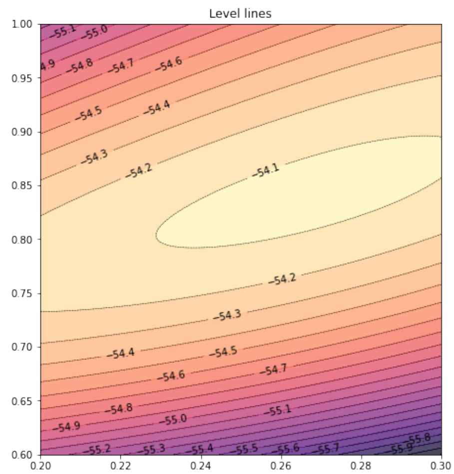
3. The table 1 contains the number of tornado occurrences in Lafayette Parish, Louisiana per year from 1950 through 2009.

1950–1959	0	0	0	1	0	0	0	1	0	0
1960–1969	1	0	0	0	1	1	0	0	0	2
1970–1979	0	0	0	0	1	3	0	2	1	0
1980–1989	1	0	0	1	0	1	0	0	2	1
1990–1999	0	1	2	0	0	1	0	1	2	0
2000–2009	0	0	3	0	2	0	1	1	3	0

Table 1: Number of tornado occurrences in Lafayette Parish, Louisiana per year from 1950 through 2009

   1. Fit a Poisson model to these data. What do you think of the quality of this model ? Compare visually the frequencies and the probabilities.
   2. Generate regularly sampled values of π in the interval $[0.2; 0.30]$ and of λ in the interval $[0.6; 1]$.
   3. Represent the function l for this sample. You should obtain this kind of picture





   4. Find the best estimators of π and λ among these sampled values.
   5. Check the quality of this model by comparing the empirical and theoretical values.

4. Estimation algorithm

   1. Compute the first and second derivative of l_n with respect to π and λ.
   2. Using the function

      `scipy.optimize.minimize`

      or by implementing the Gradient descent/Newton/BFGS algorithm (your choice), implement a function `directzip` which estimates the parameters π and λ of a ZIP distribution by maximizing the log-likelihood.

5. Provide the **goodness of fit** (GOF) using the χ^2 test statistic in order to compare the Poisson and the ZIP models

$$\Delta_{\text{GOF}} = \sum_{j=0}^{k} (O_j - E_j)^2 / E_j$$

with O_j the observed counts and E_j the expected counts.