Data Science and Social Inquiry: HW4

Yu-Chang Chen and Ming-Jen Lin

December 29, 2023

Question 1: Implementing Newton's Method

In this question, we will implement Netwon's method, which is a specific version of the gradient descent algorithm, to minimize the function

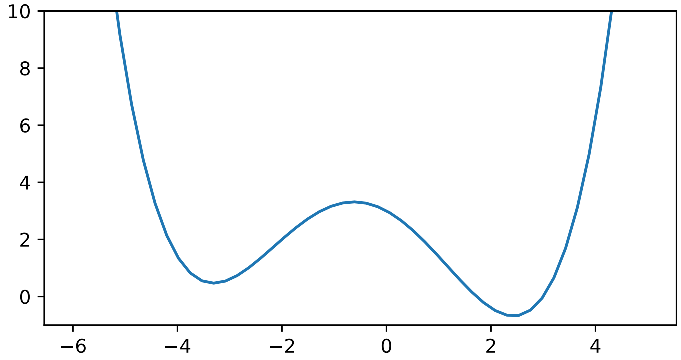
$$f(x) = 0.05x^4 + 0.1x^3 - 0.75x^2 - x + 3.$$

Recall that the Newton's method uses Hessian as learning rate and iterates in the following way

$$x_{k+1} = x_k - \frac{1}{f''(x_k)} \cdot f'(x_k).$$

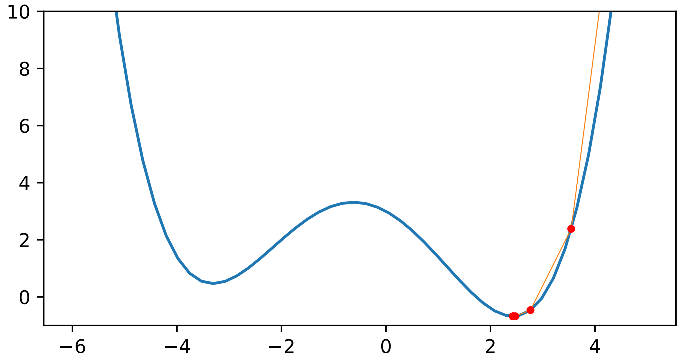
(a) (1 pt) Plot $f(x)$ in Python. Where is the global minimum?

Solution:



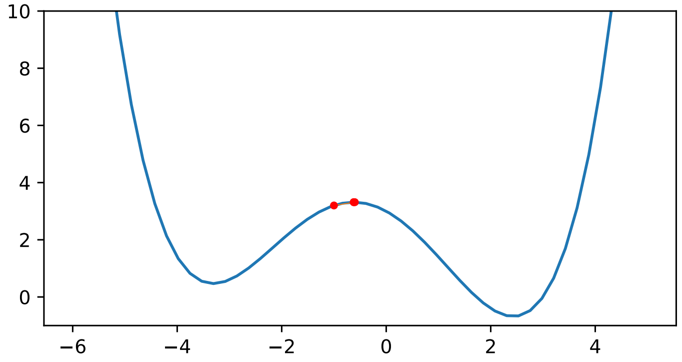
(b) (1 pt) Run the Newton's method with initial point $x_0 = 5$ and iterate 1,000 times. Plot the first 1,000 iterations on a graph. Does it converge to the global minimizer?

Solution: It converges to the global minimum.



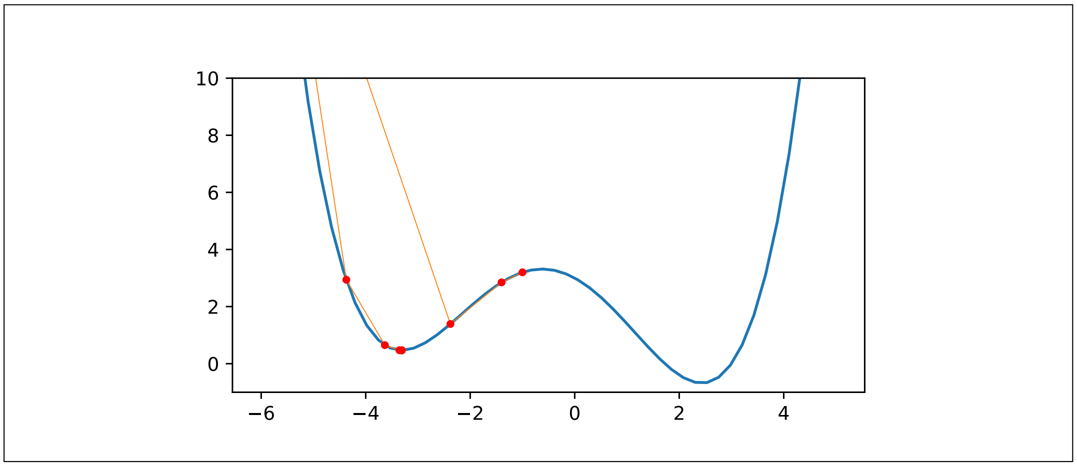
(*c*) (1 pt) Run the Newton's method with initial point $x_0 = -1$ and iterate 1,000 times. Plot the first 1,000 iterations on a graph. Does it converge to the global minimizer? Why does it behave like this? How should we fix the learning rate?

Solution:



The Hessian is required to be positive definite. In this case, $f''(-1) < 0$, so it updates in the wrong side.

One way to fix this is to 'escape' the concave zone by deliberately updating in the opposite direction. An easy way to implement the idea is to take the absolute value over Hessian.



Question 2: Apply Gradient Descent to MLE

The maximum likelihood estimator (MLE) estimates a parameter using the maximizer of the log-likelihood function. That is,

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} ln(f(x_i|\theta)),$$

where $f(x_i|\theta)$ is the p.d.f. (or p.m.f.) that generates observations $x_1, x_2, ..., x_n$.

In the case of normal distribution with known variance $\sigma^2 = 1$, the MLE of the location parameter μ is

$$\hat{\mu}_{\text{MLE}} = \arg\max_{\theta \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \left[-\frac{1}{2} ln(2\pi) - \frac{1}{2}(x_i - \mu)^2 \right]$$

Suppose that our observations x_i's are

$$3, 3, 2, 2, 4, 2, 4, 4, 3, 1.$$

Answer the following questions.

(*d*) (1 pt) Derive the Hessian of the objective function. Is it concave? [1]

[1] For maximization, we prefer concave functions since local maximum must be global maximum for concave functions.

Solution:

$$\frac{\partial \hat{\mu}_{MLE}}{\partial \mu} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu) = \bar{x} - \mu.$$

$$\frac{\partial^2 \hat{\mu}}{\partial \mu^2} = -1 < 0$$

Because the Hessian is negative, the objective function is concave everywhere.

(e) (1 pt) Analytically solve $\hat{\mu}$ by the first order condition.

Solution:

$$\hat{\mu} = \bar{x} = 2.8$$

(f) (1 pt) Use the Newton's method:

$$x_{k+1} = x_k - \frac{1}{f''(x_k)} \cdot f'(x_k)^2$$

to solve $\hat{\mu}_{\text{MLE}}$ numerically. How many iterations does it take to find $\hat{\mu}$?

Solution: At most twice update will achieve the MLE regardless of the initial choice.

Question 3: Simulating Multiple Testing

In this question, we will simulate 1,000 coins and flip each coin 100 times. Our goal is to test whether each coin i is fair or not:

$$\mathcal{H}_{i,0} : \text{Coin } i \text{ is a fair coin,}$$
$$\mathcal{H}_{i,1} : \text{Coin } i \text{ is not a fair coin.}$$

The purpose of this question is to demonstrate that, without adjustment for multiple testing, classical testing procedure may result in lots of false discovery. We'll start by constructing a "single" test for each coin.

[2] Notice that to maximize a function, we update in the direction of the gradient. But the Hessian now is negative, the sign remains minus.

Let $X_{i,1}, X_{i,2}, ..., X_{i,100} \overset{i.i.d.}{\sim} Bernouli(0.5)$ denote the 100 flips of coin i and $\bar{X}_i = \frac{1}{100}\sum_{j=1}^{100} X_{i,j}$ be their average. The Central Limit Theorem implies that

$$\bar{X}_i \overset{d}{\approx} \mathcal{N}\Big(E[\bar{X}_i],\ Var(\bar{X}_i)\Big),$$

and we can use the normal approximation to construct a t-test that rejects $\mathcal{H}_{i,0}$ if

$$|\bar{X}_i - 0.5| > c.$$

Let $\Phi(\cdot)$ be the cumulative distribution function of standard normal distribution.

(g) (1 pt) Calculate $E[\bar{X}_i]$ and $Var(\bar{X}_i)$.

Solution:

$$\mathbb{E}\left[\bar{X}_i\right] = 0.5, \quad Var(\bar{X}_i) = \frac{1}{100}(0.5)^2 = 0.0025.$$

(h) (1 pt)If we would like the test to have size 0.05, what value of the decision cutoff c should we choose? Hint: your answer will make use of $\Phi(\cdot)$.

Solution:

$$\begin{aligned} P(|\bar{X}_i - 0.5| > c) =& P\left(\left|\frac{\bar{X}_i - 0.5}{0.5/10}\right| > \frac{c}{0.5/10}\right) \\ =& 2\Phi(-\frac{c}{0.5/10}) = 0.05 \Rightarrow c = 1.96 * 0.5/10 = 0.098. \end{aligned}$$

Now, set `numpy.random.seed(13579)` and use `numpy.random.binomial()` to generate 100 × flips for 1000 coins.

(i) (1 pt)Apply the test you just constructed to test $\mathcal{H}_{i,0}$ for coins. How many false discovery, i.e., false rejections of the null hypothesis, did you find?

Solution: 61

Question 4: Solving Decisions Trees

In this question, we will solve a decision tree problem using the greedy algorithm and check whether it finds the actual global optimal solution.

Suppose we have a dataset listed in Table 1, which has 8 observations and 3 features, X_1, X_2, and X_3. Consider the following.

(*j*) (1 pt)If we only make one split, what is the best split? What is the information gain (reduction in entropy) of the best split?

Solution: The best is splitting on x_1 and the information gain is 0.1889.

$$\begin{aligned}
&E(y) = -\frac{4}{8}log_2\frac{4}{8} - \frac{4}{8}log_2\frac{4}{8} = 1\\
&E(y|x_1 = 1) = -\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4} = 0.8113\\
&E(y|x_1 = 0) = -\frac{3}{4}log_2\frac{3}{4} - \frac{1}{4}log_2\frac{1}{4} = 0.8113\\
&E(y) - E(y|x_1 = 1) - E(y|x_1 = 0) = 0.1889
\end{aligned}$$

(*k*) (1 pt)Suppose that we use the greedy algorithm to build a decision tree with two layers (i.e., two splits that result in four leaves nodes). What would the algorithm find?

Solution: The best is splitting on x_1 and then splitting on x_3.
The entropy of splitting on x_2 is 0.6887.
The entropy of splitting on x_3 is 0.5.
So we choose the lowest entropy, x_3.

(*l*) (1 pt)Generally, greedy algorithms are not guaranteed to find the global optimum. Is the solution found by the greedy algorithm in this case the global optimum? If not, find the actual global optimum by exhausting all possible splits.

Solution: The solution we find is not the global optimum. If we firt split on x_2 and then split on x_3, we can classify all data.