Optimization and Machine Learning - HW7

b04902053 鄭淵仁

June 28, 2018

Environment

- **OS:** CSIE Workstation (Archlinux)
- **Programming Language:** Matlab R2018a

Results

Timing

Gradient Descent	Newton Method	LIBLINEAR
44m41.889s	10m45.581s	21m19.126s

As you can see in the table, the Newton Method is the fastest. LIBLINEAR runs the second. And Gradient Descent is far slower than the two.

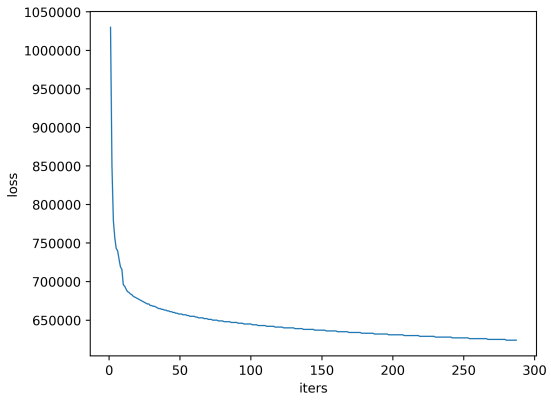
I think that's because Newton Method is written in Matlab, which has been studying the matrix computation for many years. While LIBLINEAR is written in C.

Moreover, Matlab may use parallel programming to speed up the computation, while the LIBLINEAR which I download is the single thread version. Therefore, there is no surprise that Newton Method is faster than LIBLINEAR.

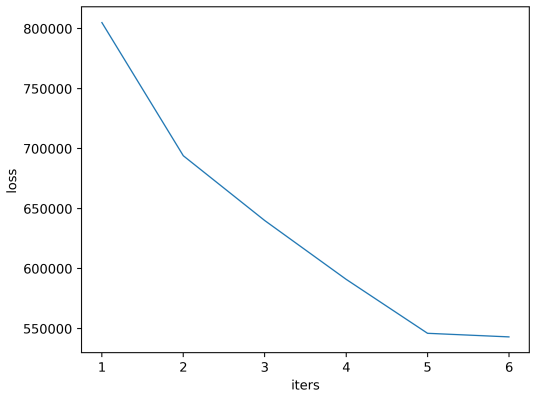
Finally, Gradient Descent does not always determining the best direction, so it is far slower than the other two method.

Loss

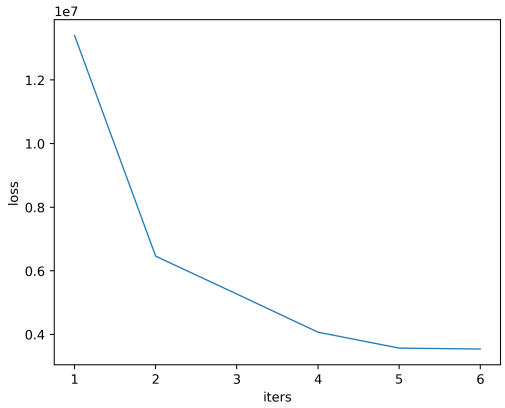
Loss of Gradient Descent



Loss of Newton Method



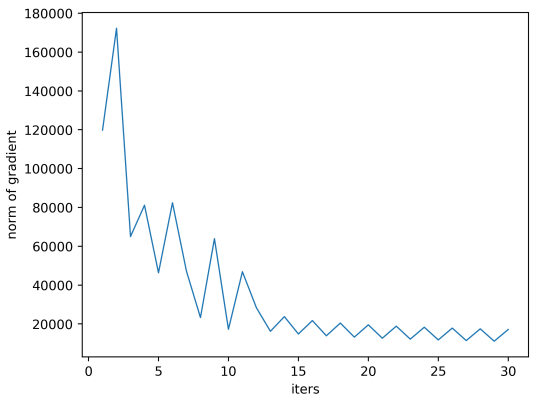
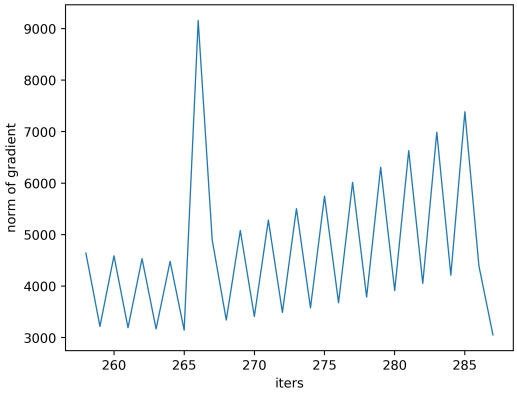
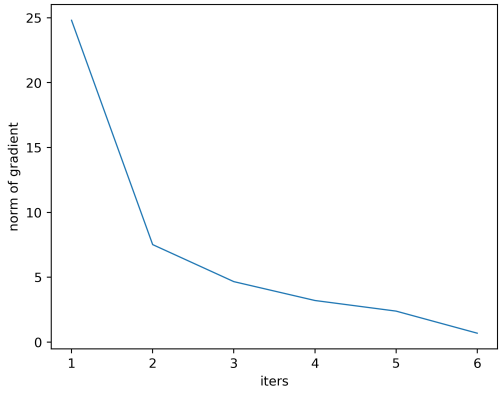
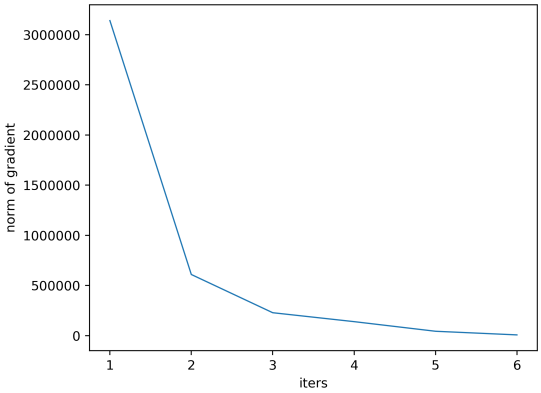
Loss of LIBLINEAR



From the above graphs, we can find that the loss of LIBLINEAR is the smallest. And that of Newton Method is larger than LIBLINEAR. While the loss of Gradient Descent is far larger than the other two.

I think that may be because that trust region causes less numerical error than conjugate gradient. Besides, my implementation may cause more errors because I am not an expert in numerical.

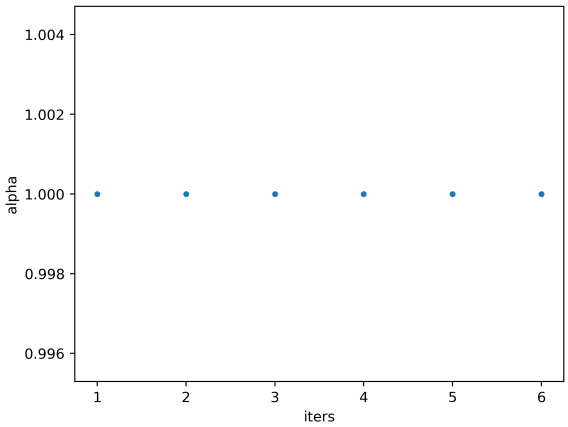
Norm of Gradient

Norm of Gradient of the first 30 iterations of Gradient Descent	Norm of Gradient of the last 30 iterations of Gradient Descent
	
Norm of Gradient of Newton Method	**Norm of Gradient of LIBLINEAR**
	

I find that the norms of gradients of the Gradient Descent do not fluently decrease, while the others decline steady.

I think this shows that Gradient Descent does not always choose a better direction in comparison to the other two methods (that is, Gradient Descent always regret the direction that it has chosen before). Therefore, Newton Method is far slower than the other two.

Alpha of Newton Method



As you can see in the graph, the alphas of Newton Method are always 1, just as the slides say.

Conclusion

1. Newton Method written in Matlab is faster than LIBLINEAR, while LIBLINEAR is more accurate than Newton Method.
2. Matlab may be faster than C because of the algorithm behind matrix computation (such as parallel programming.)
3. Trust region may cause less numerical error than conjugate gradient. And the algorithm written by me may make the error larger.
4. Gradient Descent does not always choose a better direction in comparison to the other two methods. Therefore, Newton Method is far slower than the other two.