COURSEWORK

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

477 - Computational Optimisation

Author:
Jiahao Lin (CID: 00837321)

Date: November 13, 2016

1 Part 1

1.1 Q.1

1) To prove is $\log\sum_{k=1}^{10} exp(B_{jk})$ is convex, we define:

$$f(B_j) = \log\sum_{k=1}^{10} exp(B_{jk}) \tag{1}$$

$$f(x) = \log\sum_{k=1}^{10} exp(x_k) \tag{2}$$

First we need to compute the Hessian of $f(x)$:

$$\nabla f(x) = \frac{1}{1^\top Z} \times z \tag{3}$$

$$\nabla^2 f(x) = \frac{1}{1^\top Z} \times diag(Z) - \frac{Z^\top Z}{(1^\top Z)^2} \tag{4}$$

$$where \quad Z = \sum_{k=1}^{10} exp(x_k) \tag{5}$$

For convexity we need to prove the Hessian is positive semi-definite, we need to prove:

$$\nabla^2 f(x) \geq 0 \tag{6}$$

$$v^\top \nabla^2 f(x) v >= 0 \tag{7}$$

$$v^\top \left(\frac{1}{1^\top Z} \times diag(Z) - \frac{Z^\top Z}{(1^\top Z)^2} \right) v >= 0 \tag{8}$$

$$\frac{(\sum_{k=1}^{10} Z_k V_k^2)\sum_{k=1}^{10} Z_k - \sum_{k=1}^{10}(Z_k v_k)^2}{(1^\top Z)^2} >= 0 \tag{9}$$

According to Cauchy-Schwartz inequality:

$$\sum_{k=1}^{10}(Z_k v_k)^2 \leq \sum_{k=1}^{10} Z_k \times \sum_{k=1}^{10}(Z_k V_k^2) \tag{10}$$

There for the Hessian is positive semi-definite holds, the Log-Sum-Exp function is convex.

2) Denote affine mapping on β_k as $g(\beta_k)$:

$$g(\beta_k) = x_i^\top \beta_k \tag{11}$$

And denote Log-Sum-Exp function as $f(x_k)$:

$$f(x_k) = \log\sum_{k=1}^{10} exp(x_k) \tag{12}$$

Now we know that $f(x_k)$ is convex, and we want to prove that $f \circ g$ is convex as well, from the definition of convexity, we need to prove:

$$f \circ g(\alpha x + (1-\alpha)y) \leq \alpha f \circ g(x) + (1-\alpha) f \circ g(y) \tag{13}$$
$$for \quad any \quad x, \quad y \quad and \quad \alpha \in [0,1] \tag{14}$$

Since we know:

$$g(\alpha x + (1-\alpha)y) = \alpha g(x) + (1-\alpha) g(y) \tag{15}$$

We can prove that:

$$f \circ g(\alpha x + (1-\alpha)y) = f(g(\alpha x + (1-\alpha)y)) \tag{16}$$
$$= f(\alpha g(x) + (1-\alpha) g(y)) \tag{17}$$
$$\leq \alpha f \circ g(x) + (1-\alpha) f \circ g(y) \tag{18}$$

Now we know that $\log \sum_{k=1}^{10} exp(x_i^\top \beta_k)$ is convex as well.

3) Define function $f(\beta_y)$:

$$f(\beta_{y_i}) = -x_i^\top \beta_{y_i+1} \tag{19}$$

We can show that:

$$f(\alpha \beta_{x_i+1} + (1-\alpha)\beta_{y_i+1}) = -x_i^\top(\alpha \beta_{x_i+1} + (1-\alpha)\beta_{y_i} + 1) \tag{20}$$
$$= -\alpha x_i^\top \beta_{x_i+1} - (1-\alpha) x_i^\top \beta_{y_i+1} \tag{21}$$
$$= \alpha f(\beta_{x_i+1}) + (1-\alpha) f(\beta_{y_i+1}) \tag{22}$$

Therefore $f(\beta_y)$ is convex.

4) We can define a function for $\ell 1$ Regularisation $||\beta_k||_1$:

$$||\beta_k||_1 = f(\beta_k) = \sum_{i=1}^{N} |\beta_{ki}| \tag{23}$$

To prove the convexity of $f(\beta_k)$, we can show that:

$$f(\alpha \beta_k + (1-\alpha)\beta_j) = \sum_{i=1}^{N} |\alpha \beta_k + (1-\alpha)\beta_j| \tag{24}$$
$$= \alpha \sum_{i=1}^{N} |\beta_{ki}| + (1-\alpha) \sum_{i=1}^{N} |\beta_{ji}| \tag{25}$$
$$= \alpha f(\beta_k) + (1-\alpha) f(\beta_j) \tag{26}$$

Therefore $f(\beta_k)$ is convex.

5) To prove the following function is convex:

$$\sum_{i=1}^{m}\left(\log\sum_{k=1}^{10} exp(x_i^\top \beta_k) - x_i^\top \beta_{y_i+1}\right) + \lambda \sum_{k=1}^{10} ||\beta_k||_1 \tag{27}$$

We need to prove by decomposition, in Q2, we have shown the following term to be convex:

$$\log\sum_{k=1}^{10} exp(x_i^\top \beta_k) \tag{28}$$

and in Q3, we have shown the following term is convex:

$$-x_i^\top \beta_{y_i+1} \tag{29}$$

Therefore according to sum of convex function is also convex, we can get the sum of both is convex as well:

$$\log\sum_{k=1}^{10} exp(x_i^\top \beta_k) - x_i^\top \beta_{y_i+1} \tag{30}$$

Following the same lemma, the summation term is also convex:

$$\sum_{i=1}^{m}\left(\log\sum_{k=1}^{10} exp(x_i^\top \beta_k) - x_i^\top \beta_{y_i+1}\right) \tag{31}$$

In Q4, we have shown the $\ell 1$ Regularisation term is convex:

$$||\beta_k||_1 \tag{32}$$

Then by sum of convex function is convex and affine mapping of convex function is convex, we get that the second term of optimisation problem function is convex:

$$\lambda \sum_{k=1}^{10} ||\beta_k||_1 \tag{33}$$

Combining the convex first term and convex second term, the whole optimisation problem is convex:

$$\sum_{i=1}^{m}\left(\log\sum_{k=1}^{10} exp(x_i^\top \beta_k) - x_i^\top \beta_{y_i+1}\right) + \lambda \sum_{k=1}^{10} ||\beta_k||_1 \tag{34}$$

2 Part 2

2.1 Q.2

1)

$$h(B) = \lambda \sum_{k=1}^{10} ||\beta_k||_1 \tag{35}$$

$$= \lambda \sum_{k=1}^{10} \sum_{i=1}^{n} |\beta_{ik}| \tag{36}$$

Since absolute value is not differentiable because it has a kink at 0, the whole $h(\mathcal{B})$ function is not differentiable. So $h(\mathcal{B}) \notin \mathcal{C}^1$.

2) To show the new problem is convex, we need to prove that the new regularisation term is convex. To prove the $\ell 2$ regularisation term is convex, we need to show it's Hessian is positive semi-definite,

$$||\beta_k||_2^2 = \sum_{i=1}^{10} |\beta_k|^2 \tag{37}$$

$$= \sum_{i=1}^{10} \beta_k^2 \tag{38}$$

$$\nabla(||\beta_k||_2^2) = Z \tag{39}$$

$$where \quad Z_k = 2\beta_k \tag{40}$$

$$\nabla^2(||\beta_k||_2^2) = diag(2) \tag{41}$$

Because Hessian is a diagonal matrix with all 2s on its diagonal, it's positive definite since its eigenvalues are 2 and positive eigenvalues means positive definite. Now according to sum of convex function is convex and affine mapping of convex function is convex, we proved the new regularisation term is convex:

$$\lambda \sum_{k=1}^{10} |\beta_k|_2^2 \tag{42}$$

From previous question we know the first term $g(B)$ is convex, therefore the whole new optimisation problem is convex.

$$f(B) = \sum_{i=1}^{m} \left(\log \sum_{k=1}^{10} exp(x_i^\top \beta_k) - x_i^\top \beta_{y_i+1} \right) + \lambda \sum_{k=1}^{10} ||\beta_k||_2^2 \tag{43}$$

We already now that the first term $g(B)$ is continuously differentiable, and then since the new regularisation term is continuously differentiable as well because it doesn't contain the absolute function any more, we can say that the new optimisation problem $f(B) \in \mathcal{C}^1$.

3) As shown in the code.

4) The optimization function can be expand to first order taylor series approximation:

$$f(B) \approx f(B_j) + \nabla f(B_j)^\top (B - B_j) \tag{44}$$

For the algorithm to converge, we need $\delta f(B)$ tend to be zero, which can be written as:

$$|f(B^{(j+1)}) - f(B^{(j)})| < \epsilon \tag{45}$$

On the other hand, if the second term tend to be zero, the condition will be met as well:

$$\nabla f(B_j)^\top (B - B_j) = 0 \tag{46}$$

This is corresponding to the FONC (First Order Necessary Condition):

$$\nabla f(B_j)^\top d_j \leq 0 \tag{47}$$

To satisfy the above condition, the norm of the gradient could tend to zero as well:

$$\|\nabla f(B^{(j)})\|_2 < \epsilon \tag{48}$$

Or the change for B could also tend to zero to satisfy the converge condition:

$$\|B^{(j+1)} - B^{(j)}\|_2 < \epsilon \tag{49}$$

5) Fig. 1 Shows the function values verses number of iteration.
Result for 5000 iterations using default parameters:
Func Val=141.874234; FONC Residual=2.138710; Sqr Diff=0.000214

6) The iteration of feature matrix B can be written as:

$$\beta^{j+i} = \beta^j - \alpha_j (\nabla^2 f(\beta^j))^{-1} \nabla f(\beta^j) \tag{50}$$

To find the optimal step size α_j means the minimum function value after the step, we define the problem:

$$\alpha_j = arg \quad min_{\alpha \geq 0} f(\beta^j - \alpha_(\nabla^2 f(\beta^j))^{-1} \nabla f(\beta^j)) \tag{51}$$

7) Fig. 2 Show the fast converge with secant method but not backtracking, tolerance set to 1e-8.
Result after 509 iterations converge: Func Val=141.231572; FONC Residual=0.002710; Sqr Diff=0.000000
Fig. 3 Shows no converge with secant method and backtracking, tolerance set to 1e-8.
Result after 5000 iterations with no converge: Func Val=141.242491; FONC Residual=0.512498; Sqr Diff=0.000127
When leave the algorithm to run until it converges, it stopped after 6016 iterations, results are followed:
Func Val=141.234238; FONC Residual=0.345583; Sqr Diff=0.000075

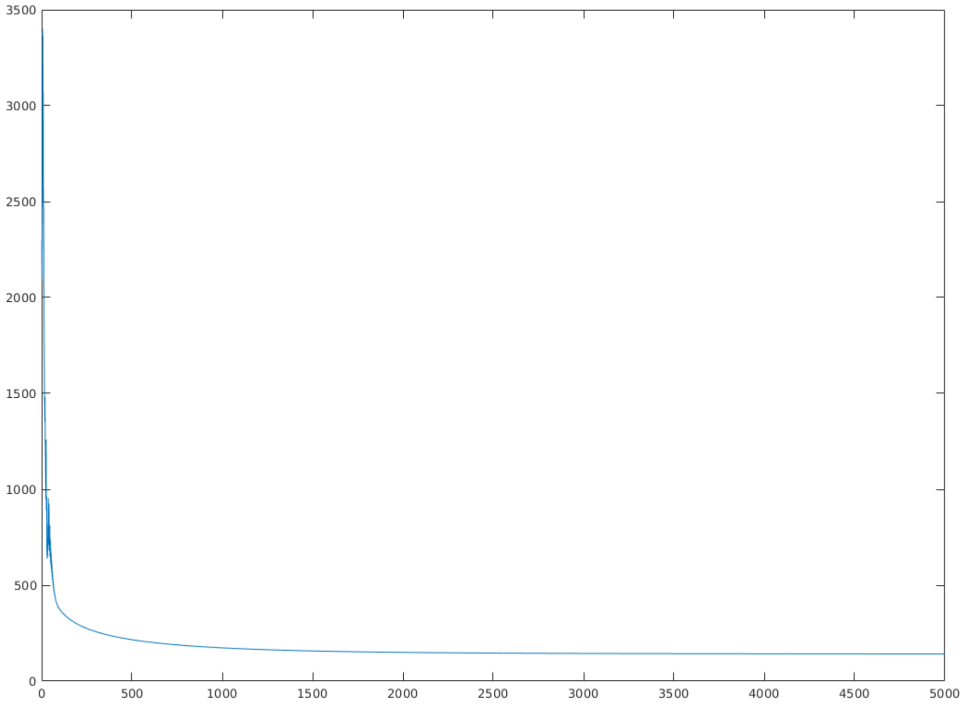

Figure 1: Function values verses number of iteration using constant step size method

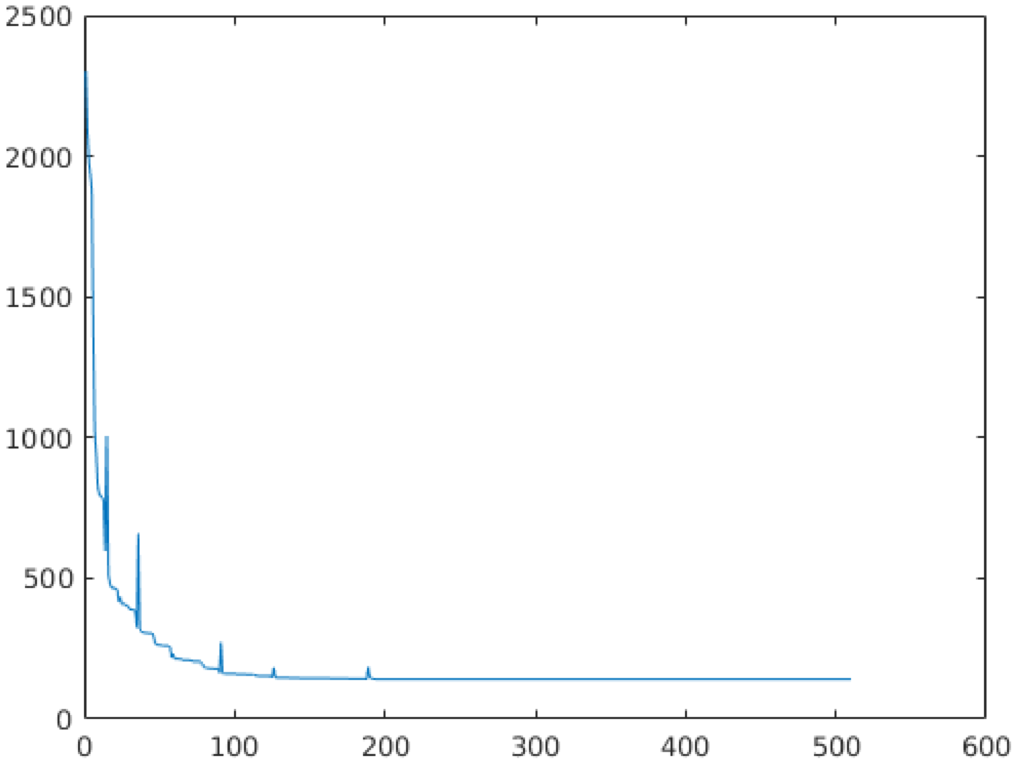

Figure 2: Secant method without backtrack

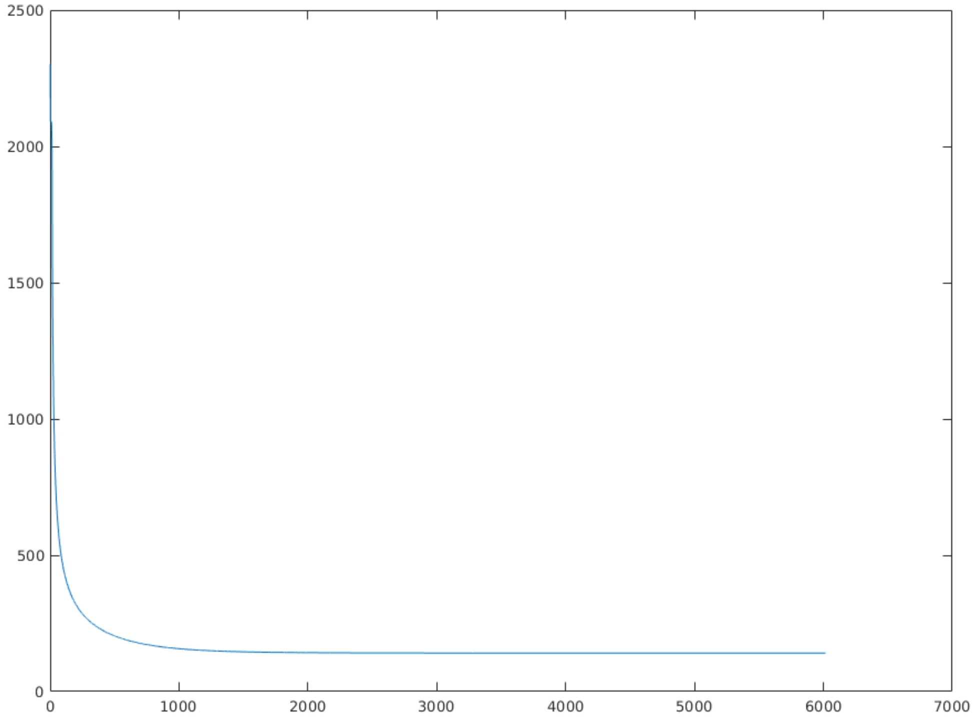

Figure 3: Secant method with backtrack