Unconstrained Optimization Algorithms

Kushal Virupakshappa
CWID: A20305737
Instructor/Professor: Yongyi Yang
T.A: Amin Zarshenas

November 5, 2018

ACKNOWLEDGEMENT

I here by acknowledge that neither any assistance has been taken from other people nor has been copied from any other source available on the internet. The sentences written in this report are my own and the work/paper/journal referred to are cited in the bibliography.

Kushal Virupakshappa
November 6, 2018

ABSTRACT

This work focuses on the implementation of the Unconstrained Optimization alogrithms. Steepest Gradient Descent Optimization, Newton's Optimiztion, BFGS Optimization and Conjugate Gradient Method Optimization methods are the specific algorithms which have been used in this work. Newton's line search method with Armijo Condition has been used to decide descent step in a given direction with the Steepest Gradient Descent and BFGS methods. For most the problem cases the convergence criteria has been met and for few problems the maximum iteration limit was reached. The convergence rate depends on the kind of optimization function and optimization algorithm.

1 KEYWORDS

Steepest Gradient Descent (SGD) Optimization, Newton's Optimization,BFGS Optimization, Conjugate Gradient Method (CGD) Optimization, Maximum Likelihood Estimation (MLE), Log Likelihood and Expectation-Maximization (EM) algorithm.

2 SOFTWARE TOOLS AND LIBRARIES

The software Tools and the libraries used in this project are listed below:

SOFTWARE TOOLS
Python, Anaconda Environment, Jupyter Notebook.

LIBRARIES
Numpy, Matplotlib.
More details regarding setup are given in readme.md file in [3].

3 PROBLEM STATEMENT

There are two parts for the project. Part 1 involves implementing SGD, Newton, BFGS and CGD algorithms as described in [1]. Part 2 involves application of the optimization algorithm to identify regions of cancerous and non cancerous pixels by estimating mean and variance for each region. The Optimization problems for each part are listed below.

3.1 Part 1 Optimization Problems

The following equations are used in optimization as given in the project problems.

$$\begin{aligned} f_{(1)}(\vec{x}) &= x_1^2 + x_2^2 + x_3^2, \\ \vec{x_0} &= (x_1, x_2, x_3)_0^T = (1,1,1)^T. \end{aligned} \tag{3.1}$$

$$\begin{aligned} f_{(2)}(\vec{x}) &= x_1^2 + 2 * x_2^2 - 2 * x_1 * x_2 - 2 * x_2, \\ \vec{x_0} &= (x_1, x_2)_0^T = (0,0)^T. \end{aligned} \tag{3.2}$$

$$\begin{aligned} f_{(3)}(\vec{x}) &= 100 * (x_2 - x_1^2)^2 + (1 - x_1)^2, \\ \vec{x_0} &= (x_1, x_2)_0^T = (-1.2, 1)^T. \end{aligned} \tag{3.3}$$

$$\begin{aligned} f_{(4)}(\vec{x}) &= (x_1 + x_2)^4 + x_2^2, \\ \vec{x_0} &= (x_1, x_2)_0^T = (2, -2)^T. \end{aligned} \tag{3.4}$$

$$\begin{aligned} f_{(5)}(\vec{x}) &= (x_1 - 1)^2 + (x_2 - 1)^2 + c * (x_1^2 + x_2^2 - 0.25)^2, \\ \vec{x_0} &= (x_1, x_2)_0^T = (1, -1)^T, c = 1, 10, 100. \end{aligned} \tag{3.5}$$

3.2 Part 2 Optimization Problem

The problem statement involves finding estimates of μ_1, σ_1^2,μ_2, σ_2^2 and P in the Probability Distribution Function(PDF) given in Eq.3.6. In Eq.3.6 i refers to the samples of the pixels in the mamography image provided in xls file.

$$\begin{aligned} p(x_i; \mu_1, \sigma_1^2, \mu_2, \sigma_2^2, P) &= P \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{\frac{-(x_i-\mu_1)^2}{2\sigma_1^2}} + (1-P) \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{\frac{-(x_i-\mu_2)^2}{2\sigma_2^2}} \\ \Theta = (\mu_1, \sigma_1^2, \mu_2, \sigma_2^2, P) &= \underset{\Theta}{argmax} (\Pi_{i=1}^{200} p(x_i; \mu_1, \sigma_1^2, \mu_2, \sigma_2^2, P)) \end{aligned} \tag{3.6}$$

According to [2], since the log likelihood Eq.3.6 does not converge a new random variable Z is introduced such that $Z = 0, 1$ and by using joint distribution and posterior probability a new algorithm called EM algorithm is introduced which reduces the optimization to maximizing Q or minimizing $-Q$ of Eq.3.7. The Expectation step of EM algorithm includes calculation Eq.3.8 using the parameters calculated in Maximization/Minimization step.

$$Q(\Theta, \Theta_{old}) = \Sigma_{i=1}^{200} \Sigma_{k=1}^{2} ((ln(P_k) + \frac{1}{2} ln(2\pi\sigma_k^2) - \frac{(x_i - \mu_k)^2}{2\sigma_k^2})(p_{ik}(z/x_i; \Theta_{old})) \tag{3.7}$$

$$p_{ik}(z/x_i; \Theta_{old}) = \frac{P_k * GaussPDF(x_{ik}; \Theta_{old})}{\Sigma_{i=1}^{200} \Sigma_{k=1}^{2} P_k * GaussPDF(x_{ik;\Theta_{old}})} \tag{3.8}$$

4 ALGORITHMS

Each of the algorithms used in this work are described in this section. SGD, Newton, BFGS, CGD, Newton's Line Search and Armijo condition algorithms are derived and detailed in [1]. EM algorithm used in application problem is described in [2].

STEEPEST GRADIENT DESCENT OPTIMIZATION METHOD

1. Initialize the parameters to be optimized and α which is the line search step.
2. Check if $\frac{||\nabla f(\vec{x})||}{1+|f(x)|}$ or Maximum Iteration reached.If yes the parameters are optimized else go to next step.
3. Compute direction vector $\vec{p}$ by using gradient of the optimization function using gradient of the function as given in $\vec{p} = -\nabla f(\vec{x})$.
4. Compute α using Newton's Line search method with Armijo Condition.
5. Compute $\vec{x_{new}} = \vec{x_{old}} + \alpha\vec{p}$.
6. Repeat step2.

NEWTON'S OPTIMIZATION METHOD

1. Initialize the parameters to be optimized.
2. Check if $\frac{||\nabla f(\vec{x})||}{1+|f(x)|}$ or Maximum Iteration reached.If yes the parameters are optimized else go to next step.
3. Compute $\nabla f(\vec{x})$ for current value of parameters.
4. Compute $\nabla^2 f(\vec{x})$,which is the hessian matrix of the given function for current value of parameters.
5. Compute $\vec{x_{new}} = \vec{x_{old}} - [\nabla^2 f(\vec{x})]^{-1} \nabla f(\vec{x})$.
6. Repeat step2.

BFGS QUASI-NEWTON OPTIMIZATION METHOD

1. Initialize the parameters to be optimized, matrix B and α which is the line search step.
2. Check if $\frac{||\nabla f(\vec{x})||}{1+|f(x)|}$ or Maximum Iteration reached.If yes the parameters are optimized else go to next step.
3. Compute direction vector $\vec{p}$ by solving the equation $B\vec{p} = -\nabla f(\vec{x})$.
4. Compute α using Newton's Line search method with Armijo Condition.

5. Compute $\vec{x_{new}} = \vec{x_{old}} + \alpha\vec{p}$.

6. Compute $B_{new} = B_{old} - \frac{(B_{old}s)(B_{old}s)^T}{(s^T B_{old}s)} + \frac{yy^T}{y^T s}$, where $s = \vec{x_{new}} - \vec{x_{old}}$ and $y = \nabla f(\vec{x_{new}}) - \nabla f(\vec{x_{old}})$.

7. Repeat step2.

CONJUGATE GRADIENT DESCENT OPTIMIZATION METHOD

1. Initialize the parameters to be optimized and direction vector to negetive ofgradient of initialized parameters i.e.,$d_0 = -\nabla f(\vec{x})$.

2. Check if $\frac{||\nabla f(\vec{x})||}{1+|f(x)|}$ or Maximum Iteration reached.If yes the parameters are optimized else go to next step.

3. Compute $\bigtriangledown^2 f(\vec{x})$,which is the hessian matrix of the given function for current value of parameters.

4. Compute α using $\alpha = \frac{-d_i \nabla f(\vec{x})}{d_i^T \nabla^2 f(\vec{x}) d_i}$.

5. Compute $\vec{x_{new}} = \vec{x_{old}} + \alpha\vec{p}$.

6. Compute β using $\beta = \frac{d_i \nabla f(\vec{x_{new}})}{d_i^T \nabla^2 f(\vec{x_{new}}) d_i}$.

7. Compute $d_{i+1} = -\nabla f(\vec{x_{new}}) + \beta d_i$

8. Repeat step2.

NEWTON'S LINE SEARCH METHOD

1. Initialize the parameters to be optimized.

2. Check if Armijo Condition is met if yes terminate with latest alpha else continue.

3. Compute $\bigtriangledown f(\alpha)$ for current value of parameters.

4. Compute $\bigtriangledown^2 f(\alpha)$,which is the hessian matrix of the given function for current value of parameters.

5. Compute $\alpha_{new} = \alpha_{old} - [\bigtriangledown^2 f(\alpha_{old})]^{-1} \bigtriangledown f(\alpha_{old})$.

6. Repeat step2.

ARMIJO CONDITION

1. $f(\vec{x} + \alpha\vec{p}) <= f(\vec{x}) + \mu\alpha\vec{p}^T \bigtriangledown f(\vec{x})$

2. If above condition is satisfied return True else return False.

Expectation Maximization Method

1. Choose an initial parameters Θ.
2. E step(Expectation step): Compute $p_{ik}(z/x_i;\Theta_{old})$ using Eq.3.8.
3. M step(Maximization or Minimization step): Minimize function $-Q(\Theta;\Theta_{old})$ given in Eq.3.7 w.r.t Θ.
4. Check for convergence if yes exit else repeat from step 2.

5 Results

The results for each optimization problem with respect of each algorithm has been provided in this section.**The screen shot of the ouput is attached in the Appendix section. Only sample examples are attached for Eq.3.1 in this section.**

Results for Optimization problem in Eq.3.1

The convergence is achieved in 1 iteration for SGD,BFGS and CGD while Newton's method achievied convergence at 19th iteration. The outputs can be seen in Figure.5.1 and Figure .5.2 for Eq.3.1.The convergence value of $\vec{x}$ is $\vec{x} = [0,0,0]^T$

Figure 5.1: Output of SGD, Newton, BFGS and CGD optimization

```
SGD Method:
Criteria of epsilon met at i:
1
[1 1 1]
[0. 0. 0.]

Newton's Method:
Criteria of epsilon met at i:
19
[1 1 1]
[0.5 0.5 0.5]
[0.25 0.25 0.25]
[0.125 0.125 0.125]
[0.0625 0.0625 0.0625]
[0.03125 0.03125 0.03125]
[0.015625 0.015625 0.015625]
[0.0078125 0.0078125 0.0078125]
[0.00390625 0.00390625 0.00390625]
[0.00195312 0.00195312 0.00195312]
[0.00097656 0.00097656 0.00097656]
[0.00048828 0.00048828 0.00048828]
[0.00024414 0.00024414 0.00024414]
[0.00012207 0.00012207 0.00012207]
[6.10351562e-05 6.10351562e-05 6.10351562e-05]
[3.05175781e-05 3.05175781e-05 3.05175781e-05]
[1.52587891e-05 1.52587891e-05 1.52587891e-05]
[7.62939453e-06 7.62939453e-06 7.62939453e-06]
[3.81469727e-06 3.81469727e-06 3.81469727e-06]
[1.90734863e-06 1.90734863e-06 1.90734863e-06]
```

Figure 5.2: Output of SGD and Newton optimization

```
BFGS Method:
Criteria of epsilon met at i:
1
[1 1 1]
[0. 0. 0.]

CGD Method:
Criteria of epsilon met at i:
1
[1 1 1]
[0. 0. 0.]
```

RESULTS FOR OPTIMIZATION PROBLEM IN EQ.3.2

The convergence is achieved in 52,54,39 and 33 iterations respectively for SGD,Newton,BFGS and CGD. The outputs can be seen in Figure.8.1 and Figure .8.2 for Eq.3.2.The convergence value of $\vec{x}$ is $\vec{x} = [1,1]^T$

RESULTS FOR OPTIMIZATION PROBLEM IN EQ.3.3

The convergence is not achieved for SGD,Newton and BFGS but for CGD convergence is achieved at 430th iteration. The outputs can be seen in Figure.8.3 and Figure .8.4 for Eq.3.3.The convergence value of $\vec{x}$ is $\vec{x} = [1,1]^T$

RESULTS FOR OPTIMIZATION PROBLEM IN EQ.3.4

The convergence is not achieved for SGD,Newton but for BFGS and CGD convergence is achieved at 42nd and379th iteration. The outputs can be seen in Figure.8.5,Figure.8.6,Figure.8.7 and Figure .8.8 for Eq.3.4.The convergence value of $\vec{x}$ is $\vec{x} = [1.5 * 10^{-2}, 0]^T$

RESULTS FOR OPTIMIZATION PROBLEM IN EQ.3.5

1. **For c=1, the output converges for SGD,Newton,BFGS and CGD at 19,124,65 and 15 iteration respectively.**The convergence value of $\vec{x}$ is $\vec{x} = [0.564, 0.564]^T$.The outputs can be seen in Figure.8.9 and Figure .8.10

2. **For c=10, the output converges for SGD,Newton,BFGS and CGD at 85,715,563 and 14 iteration respectively.**The convergence value of $\vec{x}$ is $\vec{x} = [0.4026, 0.4026]^T$.The outputs can be seen in Figure.8.11 and Figure .8.12

3. **For c=100, the output of Newton and BFGS doesn't converge while output of SGD and CGD converges at 863 and 223 iterations respectively.**The convergence value of $\vec{x}$ is $\vec{x} = [0.3597, 0.3597]^T$.The outputs can be seen in Figure.8.12 and Figure .8.13

RESULTS FOR APPLICATION(OPTIONAL) PROBLEM

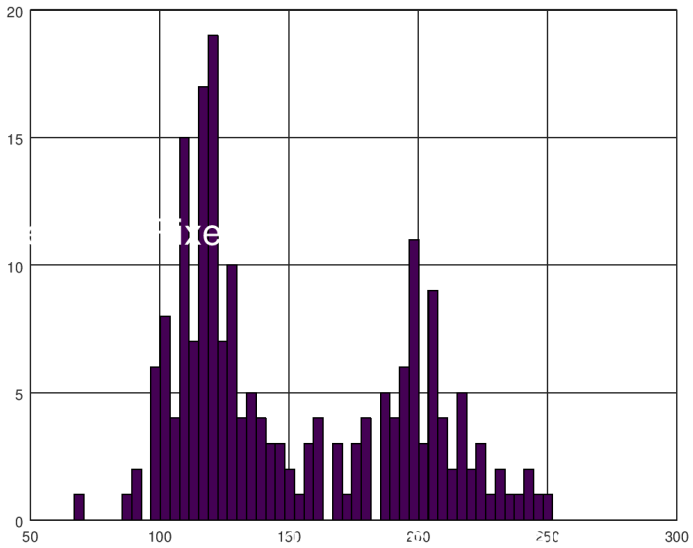
The Application problem is delt using EM algorithm with Q function optimized using Steepest Gradient Descent Method for Maximization step.The algorithm was started with seed value of $\Theta = (\mu_1, \sigma_1^2, \mu_2, \sigma_2^2, P) = (100, 50, 150, 50, 0.5)$. After 1000 iterations of EM algorithm there was no convergence but the new Θ value is $\Theta = (120.261310, 196.28040, 201.659041, 218.19920, 0.6165765)$.The output can be seen in Figure.8.15

6 EVALUATION OF RESULTS

- It can be observed that SGD decreases at linear rate for all the optimization problems.Hence slower compared to all other methods in cases where the condition of Q or Hessian matrix is comparetively smaller as in Eq.3.1,Eq.3.2 and Eq.3.3. It is still linear in Higher values for condition of Q but BFGS and Newton's method might fail to converge.

- Newton's method in theory should converge at quadratic rate but even for simple case as in Eq.3.1 it is severely limited by the precision or accuracy while in other cases there are intermediary hessian matrix which may not be singular thus triggering unwanted outputs and hence will not converge as in Eq.3.3
- BFGS method does well for most of the cases but fails to converge whenever the matrix B is not singular as in Eq.3.3. Compared to SGD it does well in most of the cases but the convergence rate slows done for the cases where condition of Q is high.
- CGD converges in all the cases and outperforms all the methods taking fewer number of steps when compared to other algorithms and matches theoretical explanation that it converges in fixed number of steps.
- Though the EM alogrithm didnot converge it has given new Θ value which is pretty close to the visual histogram representation of data as shown in Figure.6.1.

Figure 6.1: Histogram of Mammogram pixel samples



7 Conclusion

The algorithms implemented work as expected to theory but the rate of convergence is effected by accuarcy for Newton's method which downplays the strength of Newton's optimization implementation in practical scenarios.The rest of the algorithms hold up their end of the promises matching the theory for all the scenarios and having pitfalls as explained in theory and the Newton's Line Search method with Armijo condition works according to the theory and

is limited by the seed values of α and μ as there are situtions where the Armijo condition may not get satisfied,This was resolved by keeping the μ as low as possible for given optimization problem or by changing the seed value of optimizing parameters.The EM algorithm works as expected and moves toward acheiving the Maximum Likelihood Estimate and is limited only by the kind of optimization algorithm used to solve the maximization of Q function.

8 Future Work

1. Implement Secant method of Line search and Wolfe Condition to check the validity of the implemented algorithms for given optimization problems.
2. Implement EM algorithm with CGD method to optimize Q function so that convergence can be achieved.
3. Calculate the convergence rate for each algorithm using the formulas provided in the [1] and verify the convergence rate for each function and algorithm.

References

[1] Griva.I, Nash .S.G,and Sofer .A,"Linear and Non Linear Optimization," *Society for Industrial and Applied Mathematics*,2nd Ed,ISBN 978-0-898716-61-0,2009.

[2] Bishop,C.,"Pattern Recognition and Machine Learning," *Springer*,ISBN-10: 0-387-31073-8,Pg. 430-443,February 2006.

[3] Virupakshappa .K,"Github link to Project Source Code and Useage," `https://github.com/kushalviit/Unconstrained_Optimization_ECE505.git`.

Appendix

The source code can be downloaded and used/tested by using the GitHub link in [3]. Since the number of code lines is large link has been provided to Github jupyter notebook or see the python file source_code.py attached separately.

Source Code of all the four optimization algorithms

`https://github.com/kushalviit/Unconstrained_Optimization_ECE505/blob/master/project_notebook.ipynb` **Go to PART1-Algorithmic Implementation 1.All the Optimization Algorithm**

Source Code of all the four optimization algorithms

`https://github.com/kushalviit/Unconstrained_Optimization_ECE505/blob/master/project_notebook.ipynb` **Go to 2.Function 1 by scrolling down**

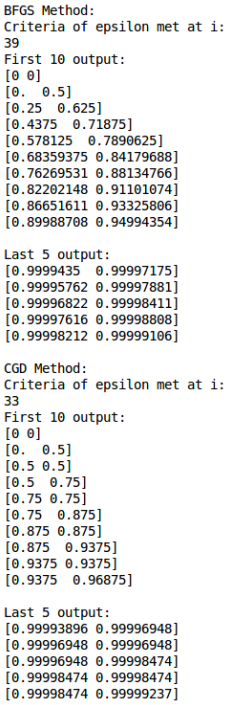
IMAGES OF OUTPUTS FOR OPTIMIZATION PROBLEM IN EQ.3.2 AND SOURCE CODE

Source Code: `https://github.com/kushalviit/Unconstrained_Optimization_ECE505/blob/master/project_notebook.ipynb` **Go to 3.Function 2 by scrolling down**

Figure 8.1: Output of SGD and Newton optimization

```
SGD Method:
Criteria of epsilon met at i:
52
First 10 output:
[0 0]
[0.  0.5]
[0.25 0.5 ]
[0.375 0.625]
[0.5    0.6875]
[0.59375 0.75   ]
[0.671875 0.796875]
[0.734375  0.8359375]
[0.78515625 0.8671875 ]
[0.82617188 0.89257812]

Last 5 output:
[0.99995528 0.99997236]
[0.99996382 0.99997764]
[0.99997073 0.99998191]
[0.99997632 0.99998537]
[0.99998084 0.99998816]

Newton's Method:
Criteria of epsilon met at i:
54
First 10 output:
[0 0]
[1.5 1. ]
[1.   0.75]
[1.125 0.875]
[1.0625 0.875 ]
[1.0625  0.90625]
[1.046875 0.921875]
[1.0390625 0.9375   ]
[1.03125    0.94921875]
[1.02539062 0.95898437]

Last 5 output:
[1.00000652 0.99998944]
[1.00000528 0.99999146]
[1.00000427 0.99999309]
[1.00000345 0.99999441]
[1.00000279 0.99999548]
```

Figure 8.2: Output of BFGS and CGD optimization

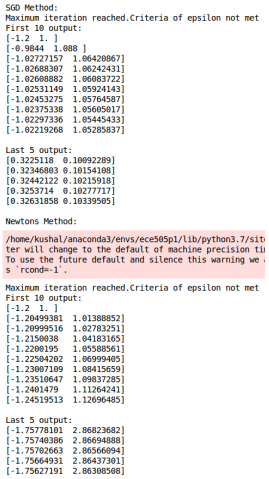


```
BFGS Method:
Criteria of epsilon met at i:
39
First 10 output:
[0 0]
[0.  0.5]
[0.25  0.625]
[0.4375  0.71875]
[0.578125  0.7890625]
[0.68359375 0.84179688]
[0.76269531 0.88134766]
[0.82202148 0.91101074]
[0.86651611 0.93325806]
[0.89988708 0.94994354]

Last 5 output:
[0.9999435  0.99997175]
[0.99995762 0.99997881]
[0.99996822 0.99998411]
[0.99997616 0.99998808]
[0.99998212 0.99999106]

CGD Method:
Criteria of epsilon met at i:
33
First 10 output:
[0 0]
[0.  0.5]
[0.5 0.5]
[0.5  0.75]
[0.75 0.75]
[0.75  0.875]
[0.875 0.875]
[0.875  0.9375]
[0.9375 0.9375]
[0.9375  0.96875]

Last 5 output:
[0.99993896 0.99996948]
[0.99996948 0.99996948]
[0.99996948 0.99998474]
[0.99998474 0.99998474]
[0.99998474 0.99999237]
```

IMAGES OF OUTPUTS FOR OPTIMIZATION PROBLEM IN EQ.3.3 AND SOURCE CODE

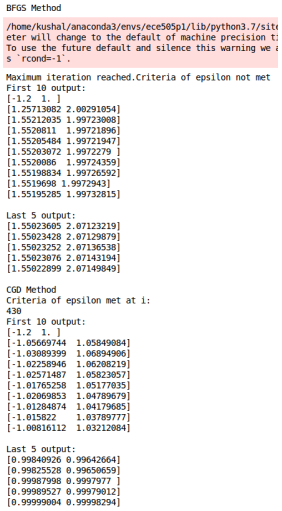
Source Code: `https://github.com/kushalviit/Unconstrained_Optimization_ECE505/blob/master/project_notebook.ipynb` **Go to 4.Function 3 by scrolling down**

Figure 8.3: Output of SGD and Newton optimization



```
SGD Method:
Maximum iteration reached.Criteria of epsilon not met
First 10 output:
[-1.2  1. ]
[-0.9844  1.008 ]
[-1.02727157  1.06420867]
[-1.02688307  1.06242431]
[-1.02608882  1.06083722]
[-1.02531149  1.05924143]
[-1.02453275  1.05764587]
[-1.02375338  1.05605017]
[-1.02297336  1.05445433]
[-1.02219268  1.05285837]

Last 5 output:
[0.3225118  0.10092289]
[0.32346803 0.10154108]
[0.32442122 0.10215918]
[0.3253714  0.10277717]
[0.32631858 0.10339505]

Newtons Method:

/home/kushal/anaconda3/envs/ece505p1/lib/python3.7/sit
ter will change to the default of machine precision ti
To use the future default and silence this warning we
s `rcond=-1`.

Maximum iteration reached.Criteria of epsilon not met
First 10 output:
[-1.2  1. ]
[-1.20499381  1.01388852]
[-1.20999516  1.02783251]
[-1.2150038   1.04183165]
[-1.2200195   1.05588561]
[-1.22504202  1.06999405]
[-1.23007109  1.08415659]
[-1.23510647  1.09837285]
[-1.2401479   1.11264241]
[-1.24519513  1.12696485]

Last 5 output:
[-1.75778101  2.86823682]
[-1.75740386  2.86694888]
[-1.75702663  2.86566094]
[-1.75664931  2.86437301]
[-1.75627191  2.86308508]
```

Figure 8.4: Output of BFGS and CGD optimization



IMAGES OF OUTPUTS FOR OPTIMIZATION PROBLEM IN EQ.3.4 AND SOURCE CODE

Source Code: `https://github.com/kushalviit/Unconstrained_Optimization_ECE505/blob/master/project_notebook.ipynb` **Go to 5.Function 4 by scrolling down**

Figure 8.5: Output of SGD optimization

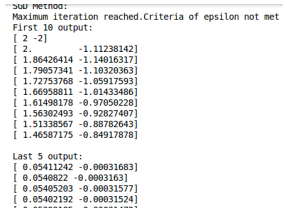


Figure 8.6: Output of Newton optimization

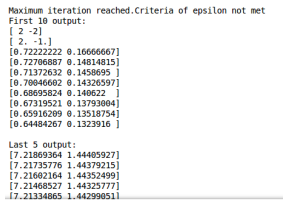


Figure 8.7: Output of BFGS optimization

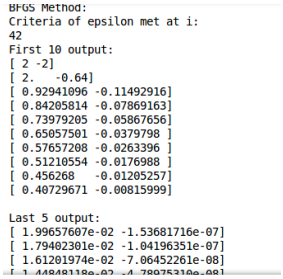
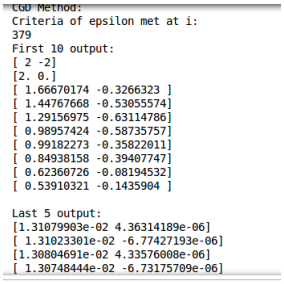


Figure 8.8: Output of CGD optimization



IMAGES OF OUTPUTS FOR OPTIMIZATION PROBLEM IN EQ.3.5 AND SOURCE CODE

`https://github.com/kushalviit/Unconstrained_Optimization_ECE505/blob/master/project_notebook.ipynb` **Go to 6.Function 5 by scrolling down**

Figure 8.9: Output of SGD and Newton optimization for C=1

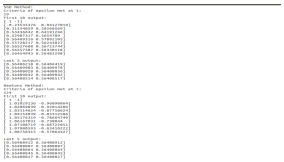


Figure 8.10: Output of BFGS and CGD optimization for C=1

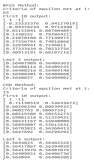


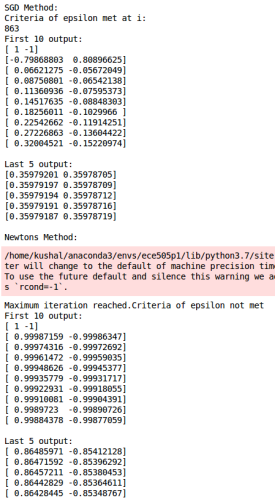
Figure 8.11: Output of SGD and Newton optimization for C=10

```
SGD Method:
Criteria of epsilon met at i:
85
First 10 output:
[ 1 -1]
[-0.91042859  1.01959594]
[ 0.88241817 -0.78292189]
[-0.86876882  0.17493908]
[-0.02654722  0.26097029]
[0.02423446 0.35292918]
[0.08079847 0.43635274]
[0.13565275 0.49239782]
[0.18122416 0.51426841]
[0.21655543 0.51421861]

Last 5 output:
[0.40260497 0.40261507]
[0.40260565 0.40261439]
[0.40260624 0.4026138 ]
[0.40260676 0.40261328]
[0.4026072  0.40261284]

Newtons Method:
Criteria of epsilon met at i:
715
First 10 output:
[ 1 -1]
[ 0.99904811 -0.9982765 ]
[ 0.99809623 -0.99654945]
[ 0.99714436 -0.99481881]
[ 0.99619252 -0.99308456]
[ 0.99524071 -0.99134668]
[ 0.99428893 -0.98960515]
[ 0.99333721 -0.98785993]
[ 0.99238553 -0.98611101]
[ 0.99143392 -0.98435836]

Last 5 output:
[0.40261049 0.40261049]
[0.40261048 0.40261048]
[0.40261046 0.40261046]
[0.40261045 0.40261045]
[0.40261044 0.40261044]
```

Figure 8.12: Output of BFGS and CGD optimization for C=10

```
Criteria of epsilon met at i:
563
First 10 output:
[ 1 -1]
[-0.91042859  1.01959594]
[-0.8836515   0.99562031]
[-0.88001838  0.98405935]
[-0.59716038  1.16074683]
[-0.59277728  1.14811976]
[-0.5869702   1.13926082]
[-0.58107768  1.13053215]
[-0.57521505  1.12190595]
[-0.56938779  1.11337778]

Last 5 output:
[0.40260756 0.40261283]
[0.40260762 0.40261275]
[0.40260769 0.40261268]
[0.40260775 0.40261261]
[0.40260781 0.40261254]

CGD Method:
Criteria of epsilon met at i:
14
First 10 output:
[ 1 -1]
[ 0.69811534 -0.68086479]
[ 0.5182517  -0.46775745]
[ 0.43278195 -0.30683507]
[0.53211281 0.14169245]
[0.47349846 0.46882753]
[0.41433481 0.41323233]
[0.4034444  0.40251673]
[0.40298153 0.40221345]
[0.40272587 0.40259758]

Last 5 output:
[0.40266179 0.40255483]
[0.40262602 0.40260834]
[0.40261716 0.40260241]
[0.40261222 0.40260979]
```

Figure 8.13: Output of SGD and Newton optimization for C=100



```
SGD Method:
Criteria of epsilon met at i:
863
First 10 output:
[ 1 -1]
[-0.79868803  0.80896625]
[ 0.06621275 -0.05672049]
[ 0.08750801 -0.06542138]
[ 0.11360936 -0.07595373]
[ 0.14517635 -0.08848303]
[ 0.18256011 -0.1029966 ]
[ 0.22542662 -0.11914251]
[ 0.27226863 -0.13604422]
[ 0.32004521 -0.15220974]

Last 5 output:
[0.35979201 0.35978705]
[0.35979197 0.35978709]
[0.35979194 0.35978712]
[0.35979191 0.35978716]
[0.35979187 0.35978719]

Newtons Method:

/home/kushal/anaconda3/envs/ece505p1/lib/python3.7/site
ter will change to the default of machine precision tim
To use the future default and silence this warning we a
s `rcond=-1`.

Maximum iteration reached.Criteria of epsilon not met
First 10 output:
[ 1 -1]
[ 0.99987159 -0.99986347]
[ 0.99974316 -0.99972692]
[ 0.99961472 -0.99959035]
[ 0.99948626 -0.99945377]
[ 0.99935779 -0.99931717]
[ 0.99922931 -0.99918055]
[ 0.99910081 -0.99904391]
[ 0.9989723  -0.99890726]
[ 0.99884378 -0.99877059]

Last 5 output:
[ 0.86485971 -0.85412128]
[ 0.86471592 -0.85396292]
[ 0.86457211 -0.85380453]
[ 0.86442829 -0.85364611]
[ 0.86428445 -0.85348767]
```

Figure 8.14: Output of BFGS and CGD optimization for C=100

```
BFGS Method:

/home/kushal/anaconda3/envs/ece505p
eter will change to the default of
To use the future default and silen
s `rcond=-1`.

Maximum iteration reached.Criteria
First 10 output:
[ 1 -1]
[-0.79868803  0.80896625]
[-0.79578635  0.8088569 ]
[-0.90324348  0.62290556]
[-0.90242128  0.62261102]
[-0.90174242  0.62218398]
[-0.90107125  0.62174509]
[-0.90040137  0.62130572]
[-0.89973219  0.62086674]
[-0.89906366  0.62042822]

Last 5 output:
[-0.47693426  0.38712272]
[-0.47666554  0.38707661]
[-0.47639695  0.3870309 ]
[-0.47612847  0.3869856 ]
[-0.47586012  0.3869407 ]

CGD Method:
Criteria of epsilon met at i:
223
First 10 output:
[ 1 -1]
[ 0.69591488 -0.69417725]
[ 0.50865067 -0.50361441]
[ 0.40716523 -0.39541478]
[ 0.36972144 -0.34503546]
[ 0.3752828  -0.29890963]
[ 0.41855431 -0.2967335 ]
[ 0.41042529 -0.26440717]
[ 0.46322189 -0.24232753]
[ 0.44538977 -0.21613861]

Last 5 output:
[0.35979179 0.35978741]
[0.35979158 0.35978739]
[0.35979156 0.35978763]
[0.35979137 0.35978761]
[0.35979135 0.35978783]
```

IMAGES OF OUTPUTS FOR OPTIMIZATION PROBLEM IN APPLICATION PROBLEM AND SOURCE CODE

`https://github.com/kushalviit/Unconstrained_Optimization_ECE505/blob/master/project_notebook.ipynb` **Go to 7.Application (Optional) by scrolling down**

Figure 8.15: Output of EM algorithm of Application Problem based on SGD method