Assignment 4

Peter LORENZ

January 10, 2017

Task

2 **appended to the end.**

5 After trying some different a. I think 0.1 matches best, where a good equilibrium between over- and underfitting can be found. The alphas for alpha = {steepest, newton, nesterov} = {0.01, 0.1, 0.01}. I started with 10e-* for each alpha and then changed the alpha for better results for one and the same data points.

6 **During the iterations, store the values $f(p^k) - f^*$ as well as $\|p^k - p^*\|$. At the end, plot the data. What do you find? How does the Nesterov compare to steepest descent in terms of convergence speed?**

Nesterov's performance was best by comparing all three algorithms. Sometimes the Newtons methods gets very close to the Nesterov's performance. In my stopping critera, I took a epslion, which works best for me with the value 0.01. If it is a very smaller number, Nesterov needs a lot more iterations (hundreds) to stop. It gains a good result, but maybe it could achieved by less iterations. So I tried with bigger epsilon. If it is too big, Nesterov stops to early.

First, the variable a I have chosen 2, which it does not seem to work well, because no output is shown after some few minutes. I changed it to 1.4 and it shows up very fast a result. p varies a bit and a I decided to make it smaller until I took 0.1 where the result looks good in the end and p is closer.

The Steepest Descent method is twice slower than the Newton's method and the Newton's method is twice slower than the Nesterov's method. After counting the iterations shown underneath.

```
/home/PycharmProjects/optimization/ocs_hw4/assignment4.py
-------------------------------------------------------
 Start Evaluation Environment
-------------------------------------------------------
Point 6 - Evaluation of the algorithms with f* and p*


Start the Steeptest Decent Method Jacobian Evaluation with an alpha_n 0.01.

Confusion Matrix:
[[ 195.    5.]
 [   8.  192.]]

Jacobian Matrix needed 48 iterations
Format: f(p)-f*, ||pk - p*||
(3713.6630918004616, 25.180494977745948)
(2338.29212746752, 23.062445761308105)
(1014.809149876789, 21.464212893963939)
(37.371979036040578, 20.555928415613529)
(75.162017312754273, 20.333592232018837)
(1042.1561364218799, 20.147152692792968)
(3470.1099033173646, 21.846233461234732)
(2148.3258860427941, 19.614803111370819)
```

```
(907.49523980898198, 17.976308393558327)
(16.632226052807461, 17.080752384342915)
(99.800403789243347, 17.021032177687662)
(932.32975026574627, 17.142781764209783)
(35.383579653761721, 16.198916562143811)
(83.930167305959316, 16.123661705617888)
(680.11606380541332, 16.187762540155855)
(-9.0834618483084597, 15.421146526879946)
(-6.7717232171026964, 15.402616035263652)
(3.1152389485258425, 15.373210322359062)
(9.5444649061904627, 15.341508316630909)
(75.226959659824473, 15.298324157626428)
(55.098239521666038, 15.177044811177034)
(357.24141740607206, 15.168558559230689)
(4.5769420436916448, 14.707809519827448)
(39.330601349739105, 14.670867506971224)
(18.638572429702599, 14.595963540544409)
(99.582327266864866, 14.554906771703113)
(25.037106037427755, 14.403429988597717)
(124.28280910087756, 14.363734131045103)
(18.160120937225464, 14.180635387577148)
(83.872257965107991, 14.140516167983929)
(12.706602726413742, 14.008634971648577)
(56.725260025164509, 13.970035891615584)
(7.0488662028038078, 13.873244830329712)
(33.835704804735379, 13.837876440747154)
(1.4704209514223692, 13.771231518126816)
(15.701171028716608, 13.740722941531558)
(-3.5997369160375925, 13.6983586608852)
(2.6588725867142529, 13.673797574438826)
(-7.6153463984300771, 13.649122461138884)
(-5.4586841611504795, 13.630459532591834)
(-10.305047490458506, 13.616775636957609)
(-9.7901915865154621, 13.602830279256693)
(-11.912658047336748, 13.594795246182494)
(-11.918996028333702, 13.584126508061086)
(-12.833314204914728, 13.578631988805128)
(-12.958790916789901, 13.570125536978228)
(-13.354710136745538, 13.565697563591579)
(-13.477380142016735, 13.558611258951947)
(-13.648936660200611, 13.554601127610924)
-- finished --


Start the Newton Method Hessian Evaluation with an alpha_n2 0.4.
newton error 0.00691084673779

Confusion Matrix:
[[ 194.    6.]
 [   8.  192.]]

Newton needed 28 iterations
Format: f(p)-f*, ||pk - p*||
(2792.0383066159716, 1460.7268833864805)
(2534.2030381373934, 1320.9314281052773)
(2297.4207682946371, 1192.7182609503516)
```

```
(2079.9223549731037, 1075.1114866523485)
(1880.0881786183827, 967.2177574950548)
(1696.4354916459793, 868.21929617759668)
(1527.6124083422783, 777.36751194737542)
(1376.3363276957898, 693.98110560900886)
(1239.312333954425, 618.11907174138457)
(1112.4834757175652, 548.4024701722002)
(995.02780318239411, 484.30156725348445)
(886.17411468814066, 425.34585384253609)
(785.21828865806208, 371.10464783857014)
(691.71282445368433, 321.18641220335201)
(606.7557214226581, 275.45154470987887)
(527.7391211073857, 234.19391875462367)
(453.88499946568362, 196.27533076429978)
(385.48124346644158, 161.57113744602631)
(322.61233778906069, 130.24561643425469)
(265.23252113891198, 102.33525301686615)
(212.95411610747291, 77.804031213675373)
(166.10313406115614, 56.753828887964886)
(123.98036883526245, 39.335663818617917)
(85.604168136338501, 24.827871226143035)
(52.637577629412526, 12.5778626337666)
(25.432649897331451, 2.0722488210458327)
(2.5478855815221593, 6.800284954702918)
(-12.279080722491408, 11.923335458001326)
(-13.50008129285969, 14.034080660818045)
-- finished --


Start the Nesterov Evaluation with an alpha 0.02.
nesterov error 0.00204703325585

Confusion Matrix:
[[ 194.    6.]
 [   8.  192.]]

Nesterov  needed 14 iterations
Format: f(p)-f*, ||pk - p*||
(7928.6905957232266, 31.082158541701578)
(5153.4084173298679, 24.867133602653148)
(1750.0558552808664, 18.584225011513229)
(3096.9975342274765, 16.305030976560886)
(4876.8146227617399, 20.541224704584653)
(6198.3260720104208, 22.514035881476225)
(4498.2739932633458, 17.223706075550719)
(1114.6973777976632, 6.9040297138116049)
(0.94086678069842122, 0.20582166577395583)
(5214.1002046137246, 20.203963941650734)
(7713.3230308747407, 29.028678308589821)
(7225.205846885452, 29.972922362607598)
(4493.8356703200834, 26.281669587458619)
(878.69176581918168, 23.156294513108477)
(41.660582516596996, 25.520777209272509)
-- finished --
```

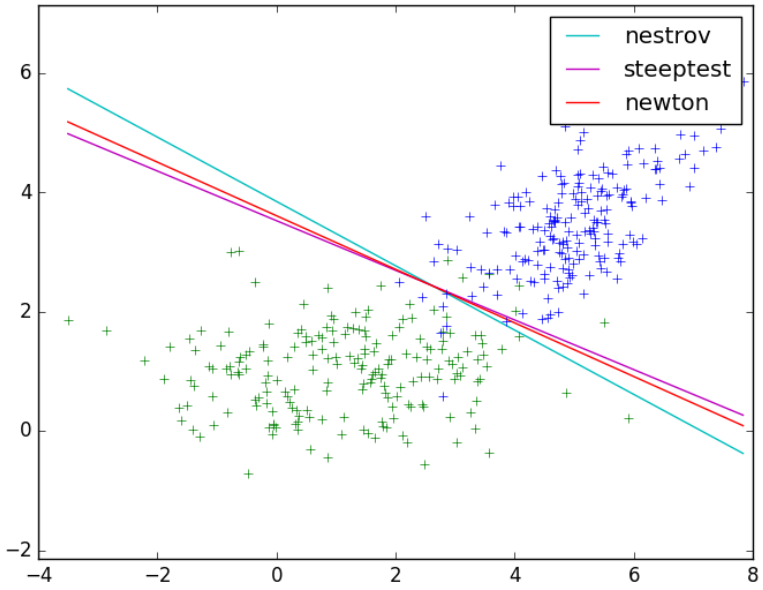

Figure 1: Graph

7 **Generate new test examples from the same Gaussian distributions and evaluate the classification error.**

Confustion Matrix [1]:

Above, I appented the iterations and the confusion matrix. The confusion matrix shows up in the main diagonal the True Positives where prediction and actual is the same. The others shows the false classifications.

All three methods shows up the pretty close classification results in the end.

[1] https://en.wikipedia.org/wiki/Confusion_matrix