10-605 Assignment 5: Stochastic Gradient Descent

Namit Katariya (andrew id: nkatariy)

March 19, 2013

QUESTION 0

- Output of "make demo" with dictionary size 10000 and μ =0.1 (cumulative log probability printed):

```
[pt,tr,hu,es,ru,pl,ca,nl,sl,fr,ga,de,hr,el] [es,fr]     -2.5163927479098613
[es,ru,ca,fr,de,el]                         [de,fr,pt]  -5.230361667820243
[pt,es,ru,pl,fr,de]                         [de,pl]     -4.52829260056261
[pt,es,ru,fr,de]                            [fr,pt]     -3.119880323675378
[es,pl,fr,de]                               [es,fr]     -3.6469902180384097
[pl,ca,de]                                  [pl]        -6.194962572700562
[pt,de]                                     [fr]        -5.204173354282212
[pt,es,pl,fr,de]                            [fr,pt]     -4.178269526803922
[es,ru,de]                                  [fr,pl]     -5.451279489139332
[fr,de]                                     []          -4.61184834723814
[nl]                                        [de,fr]     -5.268614630149776
[fr,ga]                                     [pl]        -5.176682704814612
[hu]                                        []          -5.448188469812148
[fr]                                        [de,fr,pl]  -4.00770730247661
[fr,de]                                     []          -5.751580796391452
[pt,es,pl,ca,nl,fr]                         [fr,nl,pl]  -5.9053061640367215
[el]                                        [fr]        -6.414965515545923
[de]                                        [fr]        -3.656166537983577
[fr]                                        []          -6.001685320724161
[pt,fr]                                     [fr]        -5.740815356797156
[pl]                                        [pl]        -4.1147689641279035
[ru,pl,de]                                  []          -5.770545344041741
[fr]                                        [fr,pl]     -6.208966918246572
[pt,fr]                                     [fr]        -5.5445391905239445
[sl,fr,de]                                  []          -4.987012844617406
[nl]                                        [fr]        -4.694053267689198
[ru,pl]                                     [de,fr]     -4.248049647959551
[pt,tr,es,ru,pl,ca,nl,fr,de,el]             [pl]        -5.815795450781371
[pt,es,ru,nl,fr,de]                         []          -5.323053944743495

Average acuracy = 77.83251231527095
```

-

Dataset	Average Accuracy (DSIZE=10000, μ=0.1)
abstract.tiny	77.83
abstract.small	80.74
abstract.full	80.88

QUESTION 1

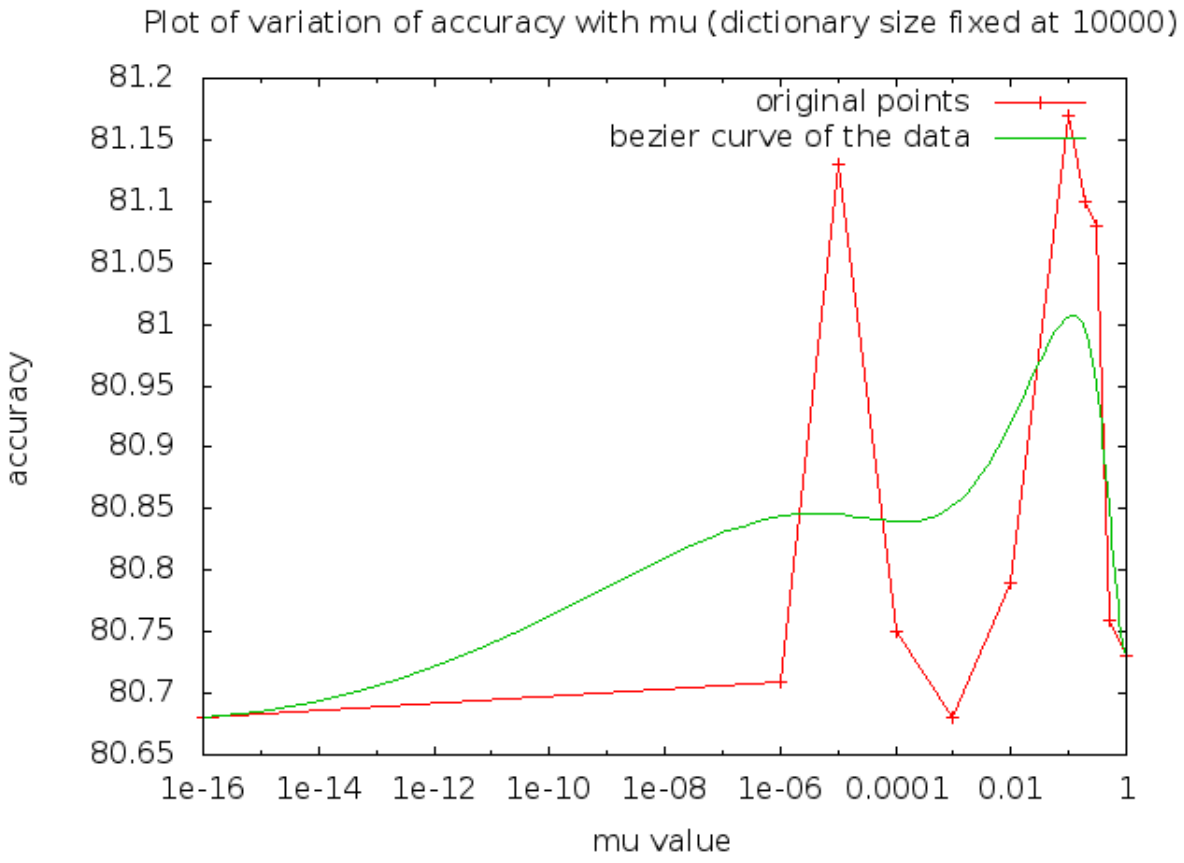
Log likelihood output for small with dictionary size 10000 and μ =0.1:

```
Iteration 1 completed. Log likelihood = -225451.68894086138
Iteration 2 completed. Log likelihood = -119668.6646817663
Iteration 3 completed. Log likelihood = -81665.64888866481
Iteration 4 completed. Log likelihood = -75234.34805401159
Iteration 5 completed. Log likelihood = -79339.44604580262
Iteration 6 completed. Log likelihood = -71771.53539579625
Iteration 7 completed. Log likelihood = -70580.47798692425
Iteration 8 completed. Log likelihood = -71707.37029522772
Iteration 9 completed. Log likelihood = -72867.84522933248
Iteration 10 completed. Log likelihood = -71147.94864523961
Iteration 11 completed. Log likelihood = -69078.08076103363
Iteration 12 completed. Log likelihood = -70697.39971563085
Iteration 13 completed. Log likelihood = -68465.81645070107
Iteration 14 completed. Log likelihood = -70064.93403673773
Iteration 15 completed. Log likelihood = -69939.81033543663
Iteration 16 completed. Log likelihood = -70439.52801302719
Iteration 17 completed. Log likelihood = -69886.82131295884
Iteration 18 completed. Log likelihood = -69350.2999505788
Iteration 19 completed. Log likelihood = -68635.54210052846
Iteration 20 completed. Log likelihood = -68808.25621767242
```

QUESTION 2

Dictionary size 10000

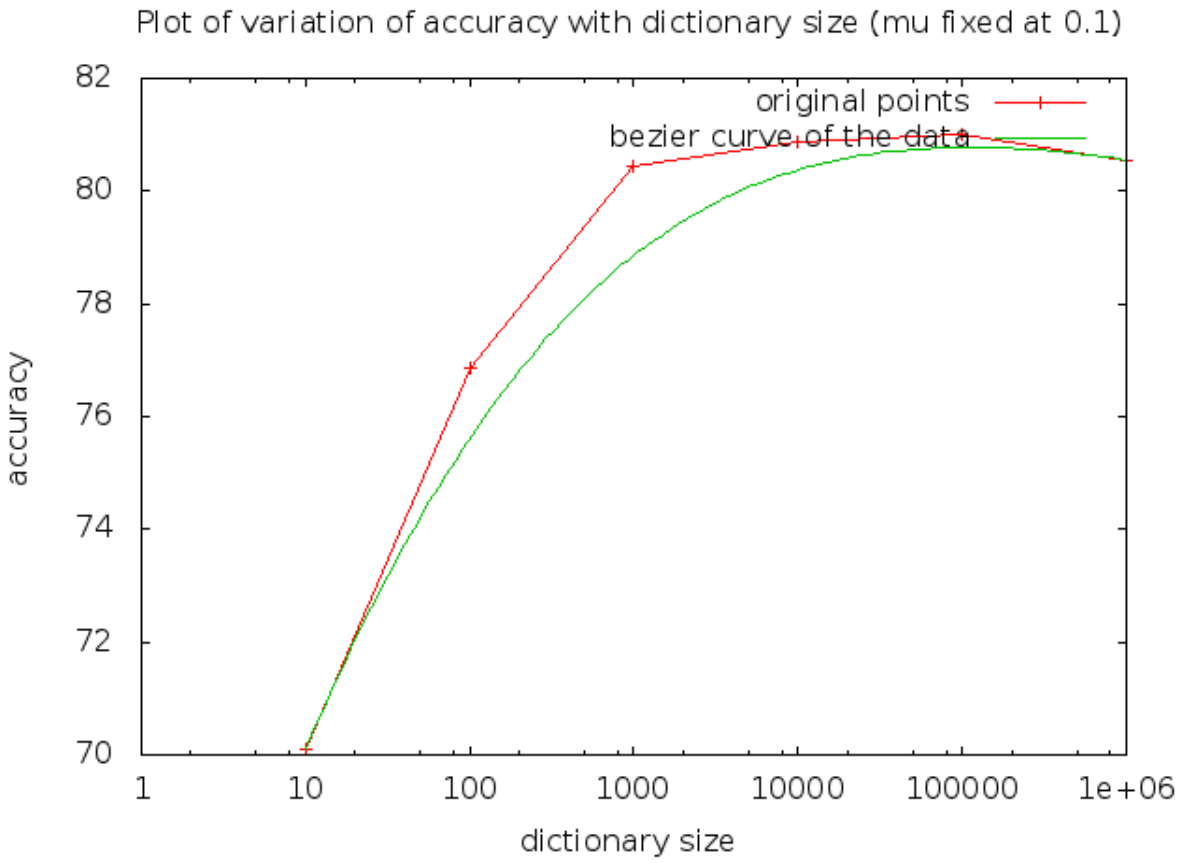
μ	0	1e-6	1e-5	1e-4	1e-3	0.01	0.1	0.2	0.3	0.5	1
Avg. accuracy	80.68	80.71	81.13	80.75	80.68	80.79	**81.17**	81.10	81.08	80.76	80.73



QUESTION 3

$\mu = 0.1$

Dictionary size	10	100	1000	1e4	1e5	1e6
Avg. accuracy	70.11	76.86	80.45	80.88	81.02	80.55



QUESTION 4

This is what I did : Along with the messages I was printing earlier, I also printed out messages for labels "NOT"+label. So for instance, if an example does not have "ca" in it true labels, the same messages will be printed out with label="NOTca". The idea is that the entire document is now effective treated as 14 documents for the binary (multi-label as before with number of labels = 2) Naive Bayes classification task for each of the 14 labels. I appropriately changed the log probability calculations and outputted average accuracy as in the current assignment.

Dataset	**NaiveBayes**	**SGD**
Tiny	40.39	77.83
Small	80.26	80.74
Full	82.41	80.88

BONUS QUESTION

- It was interesting to see the log likelihood values. They decrease upto a point but then increase, then again go down and so on. It's nice that it matches with the understanding that once you reach a trough, based on the learning rate you may sometimes overshoot and even decrease your objective (in a maximization problem)
- The average accuracy of Naive Bayes on the tiny dataset is around 40%. Since we are effectively learning 14 binary classifiers, we would expect the average to be at least better than 50% i.e the average accuracy of a purely random classifier. However, that is not the case. So this was something unexpected
- I don't know what the results would be with scaling the features to make them have similar ranges but I read online that that is important in logistic type classifiers. So I'm interested in finding out how those results would look. I did ask this question on the google group but no one responded. Also since I did not have much time, I did not try implementing what I myself suggested in the post.