CSE 446: Machine Learning (Winter 2012)
Problem Set 3: Logistic Regression

Due: **Friday, Feb 24th, 2012 at 11:59pm**

submit report and code online[1]

In this project, you will implement a logistic regression model as described in class and the reading material [1]. You may program in Python or Java. The model will be trained using gradient ascent (stop when change of conditional log likelihood $\leq$ 1e-3 or 1e-2 if the former takes a long time to reach) and tested on 4 data sets from different sources.

The data sets[2] are

- Congressional Voting Records (used in Problem Set 1)
- Wikipedia Document Classification (used in Problem Set 2)
- Pen Digits[3]
- SPECT Heart[4]

Data Format All the data sets are formatted as follows:

[label] [index1]:[value1] [index2]:[value2] ...
.
.
.

Each line contains an instance and is ended by a "\n" character (end of line). For classification, [label] is an integer indicating the class label. The pair [index]:[value] gives a feature (attribute) value: [index] is an integer starting from 1 and [value] is a real number. Indices are in ASCENDING order. Labels in the testing file should only be used to calculate accuracy or errors.

Experiments:

All experimental results need reasonable discussion.

- **Experiment A (only on the data set: Wikipedia Document Classification):**

  (2 points) Vary the learning rate $\eta > 0$. Try 5 values, 0.01, 0.05, 0.1, 0.5 and 1. Use accuracy on the validation set to determine which one works the best (in terms of accuracy; use convergence speed as tie-breaker).

  (2 points) Denote the best rate from the above, η_0. Draw a plot showing increasing conditional log likelihood during training using η_0. Let x-axis denote the iteration number. Let y-axis represent the conditional log likelihood. Draw three lines corresponding to training, validation and test respectively.

[1] `https://catalyst.uw.edu/collectit/dropbox/xling/19368`
[2] `www.cs.washington.edu/education/courses/cse446/12wi/ps3_data.tar.gz`
[3] `http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits`
[4] `http://archive.ics.uci.edu/ml/datasets/SPECT+Heart`

- **Experiment B (All data sets)**

  (4 points) Draw a plot for each data set. Vary the number of the training data by using only the first n training data points given from the data file in order to show the relative learning behavior of your naive Bayes and logistic regression classifiers.[5]. In these plots, the x-axis should represent the number of training example incremented by 1. The y-axis should represent classifier accuracy. Draw two lines, clearly labeled, one for each classifier.

Turn in the following:

- Your source code. Your code should contain appropriate comments to facilitate understanding. Include a script[6] to compile, if necessary, and run your program. The script takes **for** arguments that are (in this order) the name of the training file, the name of the validation file, the name of the test file in the same format described above and the name of output file containing the predictions of your program. For example, we should be able to execute "`./run.sh train.data validation.data test.data test.output`" to train a logistic regression model using "train.data", choose the learning rate using "validation.data", test on the file "test.data" and output predictions to "test.output". Each line of the output file has the predicted value (the integer for the class label [label]). Please make sure that the order of the output file is the same as in the test file.

  (3 points) **Evaluation Data Set:** We will test your implementation on a new data set. The performance of your implementation will be compared to a benchmark.

- (3 points) A report of at most 4 pages (letter size, 1 inch margins, 12pt font) that describes:
  - A high-level description on how your code works.
  - Implementation details that allow readers to replicate your experimental results.
  - Plots to display experimental results.
  - If you implement something for extra credit, please describe how to test your additional features.
  - If all your accuracies are low, tell us what you have tried to improve the accuracies and what you suspect is failing.

  The report will be graded based on comprehensiveness, conciseness, neatness and clarity.

Extra credit:

Feel free to extend your code in any way that you like, but to earn extra credit you also need to demonstrate how the extensions work. For example, you could repeat the experiment A for your extended code as well as for basic LR and NB. Here are just a few ideas of what you might consider trying:

- *regularization*: Impose l_2 regularization to the logistic regression model (1 point).
- *alternative learning methods*: One alternative of training a linear model is perceptron (1 point). Another option might be stochastic gradient ascent[7], the online version of gradient ascent (1 point). For perceptron, you need to use validation set to determine the number of iterations.
- *alternative numeric optimization*: Use 2nd order optimization method, e.g. conjugate gradient ascent. In addition, you need to compare the convergence speed of 1st order optimization method (i.e. gradient ascent) and conjugate gradient ascent [3].

[5] If you are unhappy with the quality of your code from problem set 2, you may compare to the implementation of naive Bayes in Weka [2]. Please describe the settings so that we can replicate your results.

[6] named as "run.sh" for Linux/Mac users and "run.bat" for Windows users.

[7] `http://en.wikipedia.org/wiki/Stochastic_gradient_descent`

References

[1] Tom Mitchell, GENERATIVE AND DISCRIMINATIVE CLASSIFIERS: NAIVE BAYES AND LOGISTIC REGRESSION. http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf

[2] Weka 3: Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/

[3] Hal Daume III, Notes on CG and LM-BFGS Optimization of Logistic Regression. 2004. http://hal3.name/docs/daume04cg-bfgs.pdf