CptS 570 Machine Learning, Fall 2016
Homework #4
Due Date: Nov 8

1. (**10 points**) Please read the following paper and briefly summarize the key ideas as you understood (You can skip the proofs, but it is important to understand the main results):

   Andrew Y. Ng, Michael I. Jordan: On Discriminative vs. Generative Classifiers: A comparison of logistic regression and naive Bayes. NIPS 2001: 841-848 `http://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf`

2. (**5 points**)

   a. Let us assume that the training data satisfies the Naive Bayes assumption (i.e., features are independent given the class label). As the training data approaches infinity, which classifier will produce better results, Naive Bayes or Logistic Regression? Please explain your reasoning.

   b. Let us assume that the training data does **NOT** satisfy the Naive Bayes assumption. As the training data approaches infinity, which classifier will produce better results, Naive Bayes or Logistic Regression? Please explain your reasoning.

3. (**5 points**)

   a. Can we compute $P(X)$ from the learned parameters of a Naive Bayes classifier? Please explain your reasoning.

   b. Can we compute $P(X)$ from the learned parameters of a Logistic Regression classifier? Please explain your reasoning.

4. (**10 points**) In the class, we looked at the log-likelihood derivation and the corresponding gradient ascent algorithm to find the parameters of a binary logistic regression classifier (see slide 12 and slide 13). We want to extend the log-likelihood derivation and parameter learning algorithm to the multi-class case. Suppose we have K different classes, and the posterior probability can be represented using the so-called soft-max function (see slide 18):

$$P(y = k|x) = \frac{exp(w_k \cdot x)}{\sum_{i=1}^{K} exp(w_i \cdot x)} \tag{1}$$

   a. Derive the log-likelihood and the corresponding gradient ascent algorithm to find the parameters.

   b. Add a regularization term to the log-likelihood objective (see slide 16), and derive the gradient ascent update rule with the additional change.

5. (**15 points**) We need to perform statistical tests to compare the performance of two learning algorithms on a given learning task. Please read the following paper and briefly summarize the key ideas as you understood:

   Thomas G. Dietterich: Approximate Statistical Test For Comparing Supervised Classification Learning Algorithms. Neural Computation 10(7): 1895-1923 (1998) `http://sci2s.ugr.es/keel/pdf/algorithm/articulo/dietterich1998.pdf`

6. (**5 points**) Please read the following paper and briefly summarize the key ideas as you understood:

   Thomas G. Dietterich (1995) Overfitting and under-computing in machine learning. Computing Surveys, 27(3), 326-327.
   `http://www.cs.orst.edu/~tgd/publications/cs95.ps.gz`

7. (**10 points**) Please read the following paper and briefly summarize the key ideas as you understood:

   Thomas G. Dietterich (2000). Ensemble Methods in Machine Learning. J. Kittler and F. Roli (Ed.) First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science (pp. 1-15). New York: Springer Verlag.
   `http://web.engr.oregonstate.edu/~tgd/publications/mcs-ensembles.pdf`

8. (**20 points**) Fortune Cookie Classifier[1]

   You will build a binary fortune cookie classifier. This classifier will be used to classify fortune cookie messages into two classes: messages that predict what will happen in the future (class 1) and messages that just contain a wise saying (class 0). For example,

   "Never go in against a Sicilian when death is on the line" would be a message in class 0.
   "You will get an A in Machine learning class" would be a message in class 1.

   Files Provided There are three sets of files. All words in these files are lower case and punctuation has been removed.

   1) The training data:

   traindata.txt: This is the training data consisting of fortune cookie messages.
   trainlabels.txt: This file contains the class labels for the training data.

   2) The testing data:

   testdata.txt: This is the testing data consisting of fortune cookie messages.
   testlabels.txt: This file contains the class labels for the testing data.

   3) A list of stopwords: stoplist.txt

   There are two steps: the pre-processing step and the classification step. In the pre-processing step, you will convert fortune cookie messages into features to be used by your classifier. You will be using a bag of words representation. The following steps outline the process involved:

   Form the vocabulary. The vocabulary consists of the set of all the words that are in the training data with stop words removed (stop words are common, uninformative words such as "a" and "the" that are listed in the file stoplist.txt). The vocabulary will now be the features of your training data. Keep the vocabulary in alphabetical order to help you with debugging.

   Now, convert the training data into a set of features. Let M be the size of your vocabulary. For each fortune cookie message, you will convert it into a feature vector of size M. Each slot in that feature vector takes the value of 0 or 1. For these M slots, if the ith slot is 1, it means that the ith word in the vocabulary is present in the fortune cookie message; otherwise, if it is 0, then the ith word is not present in the message. Most of these feature vector slots will be 0. Since you are keeping the vocabulary in alphabetical order, the first feature will be the first word alphabetically in the vocabulary.

   Implement Naive Bayes classifier with laplace smoothing and run it on the training data. Compute the training and testing accuracy.

9. (**20 points**) Empirical analyis question. You will use the Weka: `http://www.cs.waikato.ac.nz/ml/weka/downloading.html` software. Please use the two datasets from HW-3 for this question.

   a. Bagging (weka.classifiers.meta.Bagging). You will use decision tree as the base supervised learner. Try trees of different depth (1, 2, 3, 5, 10) and different sizes of bag or ensemble, i.e., number of trees (10, 20, 40, 60, 80, 100). Compute the training accuracy and testing

[1] Thanks to Weng-Keen Wong and his advisor Andrew Moore for sharing the data.

accuracy for different combinations of tree depth and number of trees; and plot them. List your observations.

b. Boosting (weka.classifiers.meta.AdaBoostM1). You will use decision tree as the base supervised learner. Try trees of different depth (1, 2, 3) and different number of boosting iterations (10, 20, 40, 60, 80, 100). Compute the training accuracy and testing accuracy for different combinations of tree depth and number of boosting iterations; and plot them. List your observations.