

# ICS 274 Homework 4

Probabilistic Learning: Theory and Algorithms, ICS 274, Winter 2009

Due Date: Thursday, February 26th, in class

## Implementing and Testing a Logistic Regression Classifier

In this week's homework you will write MATLAB code<sup>1</sup> for a logistic regression classifier and test it on simulated and real data. Specifically:

### Logistic Regression Function in MATLAB

You are to write a MATLAB function called `logistic_train.m` that takes an input data set, a set of binary training labels, and an optional argument that specifies the convergence criterion, and returns a set of logistic weights. Specifically the function should have the following form:

```
function [weights] = logistic_train(data,labels,epsilon,maxiterations)
% [weights] = logistic_train(data,labels,epsilon,maxiterations)
%
% code to train a logistic regression classifier
%
% INPUTS:
%   data = n x (d+1) matrix with n samples and d features, where
%         column d+1 is all ones (corresponding to the intercept term)
%   labels = n x 1 vector of class labels (taking values 0 or 1)
%   epsilon = optional argument specifying the convergence
%             criterion - if the change in the absolute difference in
%             predictions, from one iteration to the next, averaged across
%             input features, is less than epsilon, then halt
%             (if unspecified, use a default value of 10^-5)
%   maxiterations = optional argument that specifies the
%                   maximum number of iterations to execute (useful when
%                   debugging in case your code is not converging correctly!)
%                   (if unspecified can be set to 1000)
% OUTPUT:
%   weights = (d+1) x 1 vector of weights where the weights
%             correspond to the columns of "data"
```

---

<sup>1</sup>Note: if you wish to use R instead, let me know if you have not talked to me already about this.

The classifier should be trained using the Newton-Raphson (IRLS) iterative procedure described in class and in the text, using the log-likelihood objective function. You can initialize all the weights at 0. You can use the MATLAB function `inv.m` to invert the Hessian matrix directly - it should work fine for the data set we are using below.

**Important Note:** If you are unable to get your logistic regression function working in time, as a backup you can look at (or even use if you wish) the logistic regression code from Professor Geoff Gordon at CMU (link on the Web site). If you use this code to help you in this homework (whether using the code directly, or just looking at it before you write your own code), then you must state so clearly in your homework submission. 30 points (out of the maximum 100) will be deducted automatically—but it may be worth doing this if you can't get your own logistic code to work. Note that this option is being provided as a backup so that you can do the rest of the homework—you should first try to write your own logistic code. It is only about 10 lines or so in MATLAB.

## Testing on Spam Email Data

On the class Website is a pointer to a directory containing a data set on spam email. There are 57 features and 2 class labels. Please read the file `spambase.DOCUMENTATION` before you use this data. There are two versions of the feature (attribute) data: `features.txt` and `binary_features.txt`, where the 2nd is the same as the first except that all features have been converted from counts into binary features (by splitting above and below the mean count). I would like you to report results using the binary version in your experiments below (you can try the “unbinary” data if you wish and report additional results in an Appendix (optional)).

You are to do the following:

1. Train and test a logistic regression model on all of the data.
  - Show a plot of the classification accuracy (on all of the data) as a function of the number of iterations of the classifier (x-axis is number of iterations, y-axis is the accuracy as a percentage).
  - Report the final classification accuracy at convergence (as a percentage).
  - Examine the weights after learning (and look at the names of the corresponding features in the file `spambase.names`), and report on 5 features whose weight values you find intuitive and informative—provide the feature names, the corresponding weight values, and why you think the weight values are informative. Note that in principle one has to be careful in interpreting weights in models such as logistic regression, since if there are 2 variables that are highly correlated then their weights may only be interpretable when looked at together. Nonetheless you should be able to find 5 variables in this data whose weights agree with what we might expect.
2. Now generate results using 10-fold cross-validation where the first test set is the first 10% of the data and you train on the remaining 90%; the second test set is the 2nd 10% of the data, and you train on the remainder; and so on until you get to the last 10% to test on. Since there are 4601 data points in total, the first 9 test sets can have 460 points and the last test set can have 461. Create your test sets in the order of the data points in the file, i.e., the first test set will be rows 1 to 460, and so on. I have already randomized the rows (and

corresponding labels) before giving you the data so that the rows are in random order (which is important for cross-validation experiments). Report the 10 accuracies that you get across the 10 test sets, and the average of the 10 accuracies. Compare to what you got in part 1 and comment briefly.

3. Create a separate test data set consisting of all rows in the file from row 2001 to 4601 inclusive (and corresponding labels). You now have 2 data sets, a training data set with 2000 rows (the first 2000 rows of the original file) and a test data set with 2601 rows. Train your logistic regression classifier on the first  $n$  rows of the training data,  $n = 10, 20, 50, 100, 200, 500, 1000, 2000$  and report the accuracy on the test data as a function of  $n$ . If you run into numerical problems (e.g., the inverse of the Hessian can't be computed for numerical reasons) then move on to the next  $n$  value (but it should work for all these values of  $n$ ). Comment on your results.

Note that you should use vectors and matrices where possible in MATLAB to make your code more efficient and also more compact. As an example, here is a simple script to read in the data, call a logistic model, make predictions, and calculate accuracies:

```
load labels.txt; y = labels;
load binary_features.txt; data = binary_features;

[n d] = size(data);

X = [data ones(n,1)];

w = logistic_train(X,y);

phat = 1 ./ (1+exp(-X*w));
chat = phat>0.5;

accuracy = 100*sum(y==chat)/length(y)
```

## What to Submit

Submit both MATLAB code (the code for the function for training the logistic classifier) and a brief report (in Word or PDF format) to the Homework 4 folder in EEE. (for students without access to EEE, please email me a Zip file with "CS 274A: Homework 4" in the subject line of your email).

## Optional Additional Problems

If you wish you may try 1 or more of the problems below and include them as an Appendix in your report. There will be no extra credit given for these problems, but if you have time you may want to try them.

### Compare Logistic and Gaussian Classifiers

Conduct a small experimental study of your logistic classifier with the Gaussian classifier you built for the last homework. You can use the spam email data set, or use other data sets (e.g., see the UCI Machine Learning Repository for data sets). You could look at whether the Gaussian or logistic tend to give better classification results in general (usually the logistic works better, unless the data happens to match well the Gaussian assumptions of the Gaussian model).

### Testing on Data used in Prior Research

Find a data set in a published paper (can be real or simulated) where results with one or more particular types of classifier were reported on the data set (e.g., using an SVM method, a decision tree, a nearest-neighbor method, or some other form of classifier) but logistic regression was **not** included in the classifiers tried. Download the data (so it must be a paper where the data is available publicly) and run logistic regression, using a similar training and test setup as used in the published paper (or something reasonably close). Compare the results you get with logistic regression (e.g., in terms of accuracy on test data) compared with the reported results in the paper. Clearly describe your experiments and provide some comments on the results you obtain.