Problem Set 7
Due on Friday, December 2, 2016 at 11:55 pm

How to Submit

Create one zip file (.zip) of your code and plots and submit it via `ilearn.ucr.edu`. This zip file should have `ps7.m` plus any other code that you have written. It should also contain your answers to the second question in `ans.txt` or `ans.pdf`.

Do not supply any directories in your zip file. Each file should (in comments) list

- Your name
- Your UCR student ID number
- The date
- The course (CS 229)
- The assignment number (PS 7)

Spam Ensemble Classifiers, Redux [4 points]

For this question, perform exactly the same experiment as in part c of PS 6 (that is, train bagging and boosting on the spam data set). However, do this only for a tree depth of 2. Plot the testing error only against the number of trees. Select the number of trees as `floor(logspace(0,2+log10(5),10))` (10 points, evenly logarithmically spaced between 1 and 500).

Now, for each of bagging and boosting separately, use L1-regularized logistic regression to retrain the weights associated with the final 500 trees. Vary λ (the regularization strength) as `logspace(-4,-1,10)`. Plot the testing error for each of these lambdas against the number of non-zero weights on the same plot as above. All told, there will be four plots (bagging, bagging reweighted, boosting, and boosting reweighted) of testing error versus number of trees.

Call the code that performs this `ps7.m`

To fit L1-regularized logistic regression, use the following code

```
% lambda is a lambda value, or a vector of lambda values
[w,other] = lassoglm(X,Y==1,'binomial','Standardize',0,'Lambda',lambda);
w = 2*w; % weight vector, or weight matrix (one column for each lambda value)
w0 = 2*other.Intercept-1; % bias term or vector of same (one for each lambda)
```

The `lassoglm` function is in the stats toolbox and you may use this one function (in the way shown above) for this problem set.

Negative Binomial Regression [5 points]

The negative binomial distribution described the number of "successes" before r "failures" if the probability of success is q:

$$p(y) = \binom{y+r-1}{y} q^y (1-q)^r$$

If r is fixed (not a free parameter), this is a member of the exponential family.[1]

[1] A reminder: you are required to do this problem without external resources. If you are stuck, please ask for help from the instructor or TA; we would be glad to help. Do not look up help online; you will only understand GLMs by trying them yourself.

part a. [2 point]

Demonstrate it is a member of the exponential family with a derivation that demonstrates $h(y)$, $\phi(y)$, and $A(\theta)$. Also, show how θ is related to the q parameter above.

part b. [2 points]

If we have sellers who are given r units to sell and we record how quickly they run out of their inventory (y), then if we take a "failure" to be the sale of an item and a "success" to be a non-sale, we might expect the distribution to be a negative binomial. We would like to predict a seller's non-sale rate (q) from other variables (x).

So, given a database of seller's information ($\{x_i\}$) and the corresponding amount of time it takes them to sell out of their inventory ($\{y_i\}$), we'd like to find a general relation. Using a generalized linear model with the negative binomial distribution would make sense.

To do so, what is the canonical link function, $g(\mu)$?

Given learned parameters w and a new point x, how would you predict q?

part c. [1 point]

What would be the Newton-Raphson update equations for learning from this type of data?