B14 : INFERENCE : HILARY 2016-2017

FRANK WOOD

Questions :

(1) Derive the maximum likelihood estimator $\hat{\lambda}$ for the Poisson distribution

$$p(x|\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

(2) Derive the first two moments of the population sampling distribution of $\hat{\lambda}$

(3) Simulate the estimator sampling distribution to perform the following hypothesis testing task. There is a radioactive emitter which emits particles at some unknown rate. Any actual rate of less than 10 particles per second is considered safe, i.e. not radioactive. Assume that you take 10 independent measurements of 5, 6, 25, 12, 4, 8, 9, 14, 7, and 9 particles per second respectively. Is the source radioactive? Use a test whose null hypothesis H_0 is "the emitter emits 10 or fewer particles per second."

(4) Given data $\{(x_1 = -1, y_1 = -5), (x_2 = 0, y_2 = 0), (x_3 = 1, y_3 = 5)\}$ derive and fit an estimator for a 3rd-order polynomial regression curve that as $x \to -\infty$ has $y \to \infty$

(5) Use gradient methods to fit maximum likelihood parameters to a logistic regression classifier of spam or not spam based on email features. There is a zip that contains a file spambase.data of email features and labels available to download from

`http://www.robots.ox.ac.uk/~fwood/teaching/B14/data/spambase.zip`

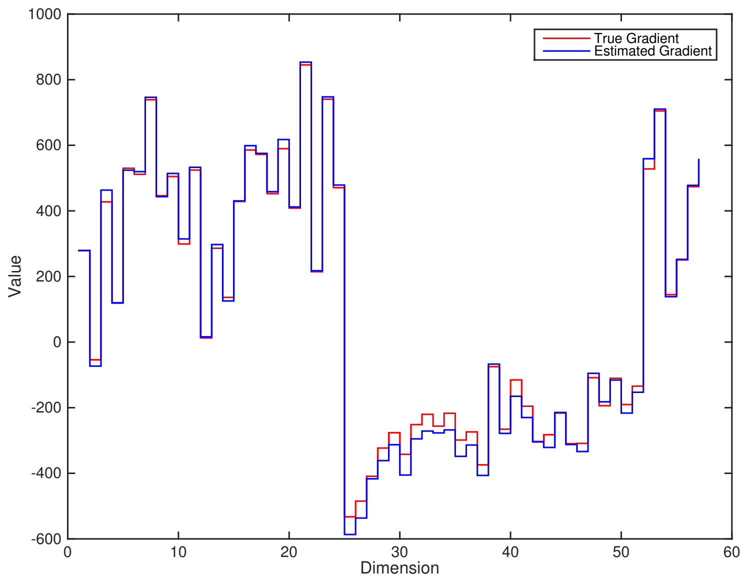
Using this data code up and train a logistic regression classifier using gradient methods to find a maximum likelihood estimate of the weight vector parameters $\boldsymbol{\beta}$. Remember that in a logistic regression classifier

$$P(y_i = 1|\mathbf{x}_i) = \frac{1}{1 + e^{-\mathbf{x}_i^T \boldsymbol{\beta}}}$$

where y_i is the $\{0, 1\}$ label and $\mathbf{x}_i$ the features of datapoint i.

You may find that verifying and using the following helps when deriving the gradient $\sigma(a) = 1/(1 + e^{-a}) \implies \frac{d\sigma}{da} = \sigma(a)(1 - \sigma(a))$.

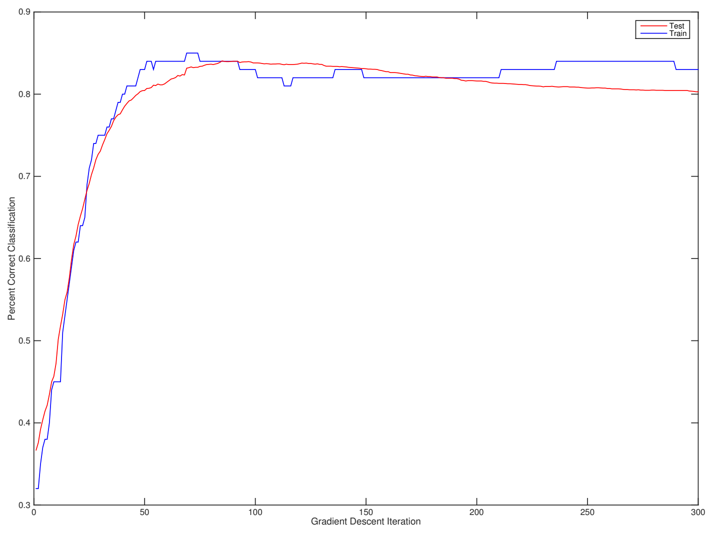
During the coding of your answer you should check your gradient calculation. Hint: you can do so by using the pre-limit definition of gradient and computing an approximation to each derivative in terms of a small perturbation. If you do this you should end up with a figure like this that shows the analytic gradient is in fact equal to the true gradient.



Your matlab code should start with

```
load spambase.data

all_labels = spambase(:,end);
all_data = spambase(:,1:(end-1));
all_data = zscore(all_data);

rand_inds = randperm(length(all_labels));
all_labels = all_labels(rand_inds);
all_data = all_data(rand_inds,:);

training_data_inds = 1:100;
test_data_inds = setdiff(1:length(all_labels),training_data_inds);

training_data = all_data(training_data_inds,:);
training_labels = all_labels(training_data_inds);

test_data = all_data(test_data_inds,:);
test_labels = all_labels(test_data_inds);
```

and you should produce a plot like



that shows classification performance on both the test and training sets as a function of gradient step iteration.

Besides submitting and demonstrating running code, questions you should answer include:

- What learning rate did you use and why?
- Why are the test and training accuracies different?
- Is there an effect in terms of which training data you use and why?