Homework Assignment 3

Due in class, Thursday October 15

SDS 383C Statistical Modeling I

1 Ridge regression and Lasso

1. Get the Prostrate cancer data from `http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/prostate.data`. More information about this dataset can be found in `http://statweb.stanford.edu/~tibs/ElemStatLearn/datasets/prostate.info.txt`.

   (a) (2.5 pts) (In class we learned about Ridge regression with tuning parameter λ. Define

   $$\mathrm{df}(\lambda) = \mathrm{tr}\left(\boldsymbol{X}(\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X} + \lambda \boldsymbol{I})^{-\lambda}\boldsymbol{X}^{T}\right).$$

   Plot the coefficients of the covariates as λ is increased from 0 to 1000. A similar plot can be found in figure 3.8 in H-T-F. This figure essentially plots the ridge regression coefficients of the covariates as dfλ is increased.

   (b) (2.5 pts) Now plot the coefficients learned by Lasso as λ is increased from 0 to 100. For this you can use the LARS package.

   (c) (5 pts) Finally reproduce columns 4 and 5 for Ridge regression and Lasso in Table 3.3. Remember to reproduce the test set prediction errors as well.

2 Discriminative vs. Generative Classifiers

A very common debate in statistical learning has been over generative versus discriminative models for classification. In this question we will explore this issue, both theoretically and practically. We will consider Naive Bayes and logistic regression classification algorithms.

To answer this question, you might want to read: *On Discriminative vs. Generative Classifiers: A comparison of logistic regression and Naive Bayes*, Andrew Y. Ng and Michael Jordan. In NIPS 14, 2002. `http://www.robotics.stanford.edu/~ang/papers/nips01-discriminativeg` `pdf`

2.1 Logistic regression and Naive Bayes

(a) (3 pts) `The discriminative analog of Naive Bayes is logistic regression`. This means that the parametric form of $P(Y|X)$ used by Logistic regression is implied by the assumptions of a Naive Bayes classifier, for some specific class-conditional densities. In the reading you will see how to prove this for a Gaussian naive bayes classifier for continuous input values. Can you prove the same for binary inputs ?Assume X_i and Y are both binary. Assume that $X_i|Y = j$ is Bernoulli(θ_{ij}), where $j \in \{0, 1\}$, and Y is Bernoulli(π).

2.2 (1+2+3+1 pts) Double counting the evidence

(a) Consider the two class problem where class label $y \in \{T, F\}$ and each training example X has 2 binary attributes $X_1, X_2 \in \{T, F\}$. How many parameters will you need to know/evaluate if you are to classify an example using the Naive Bayes classifier?

Let the class prior be $P(Y = T) = 0.5$ and also let $P(X_1 = T|Y = T) = 0.8$ and $P(X_1 = F|Y = F) = 0.7$. , $P(X_2 = T|Y = T) = 0.5$ and $P(X_2 = F|Y = F) = 0.9$. So, attribute X_1 provides a slightly stronger evidence about the class label than X_2.

i. Assume X_1 and X_2 are truly independent given Y. Write down the Naive Bayes decision rule.

★ **SOLUTION:** Given an example with attribute values (x_1, x_2): $Y = T$ if $\frac{\log P(Y=T)}{\log P(Y=F)} + \frac{\log P(X_1=x_1|Y=T)}{\log P(X_1=x_1|Y=F)} + \frac{\log P(X_2=x_2|Y=T)}{\log P(X_2=x_2|Y=F)} > 0$ else $Y = F$.

ii. Show that if Naive Bayes uses both attributes, X_1 and X_2, the error rate is 0.235. Is it better than using only a single attribute (X_1 or X_2)? Why ? The error rate is defined as the probability that each class generates an observation where the decision rule is incorrect.

★ **SOLUTION:** The expected error rate is the probability that each class generates an observation where the decision rule is incorrect: if Y is the true label, let $\tilde{Y}(X_1, X_2)$ be the result of classification (predicted class label), then the expected error rate is

$$E_e = \sum_{i\in\{T,F\}} \sum_{j\in\{T,F\}} \sum_{k\in\{T,F\}} P(X_1 = i, X_2 = j, Y = k) * I[k \neq \texttt{NB-Prediction}(X_1 = i, X_2 = j)] \tag{1}$$

where $I[z]$, is the indicator function. If condition z is *true*, then $I[z] = 1$, else $I[z] = 0$.

Given X_1, Naive Bayes will make the following predictions: $X_1 = T \Rightarrow Y = T$, if $X_1 = F \Rightarrow Y = F$. We only need to calculate $P(X_1 = F, Y = T)$ and $P(X_1 = T, Y = F)$, since this are the 2 cases when NB makes wrong prediction. In other 2 cases, when NB does not make mistake, the value of indicator function $I[z]$ from equation 1 will be 0 and the whole term will be 0. We obtain:

$$\begin{aligned} E_e = \quad & P(X_1 = F, Y = T) + P(X_1 = T, Y = F) = P(Y = T)P(X_1 = F|Y = T) \\ & + P(Y = F)P(X_1 = T|Y = F) = 0.5 * 0.2 + 0.5 * 0.3 = 0.25 \end{aligned}$$

Similarly, for X_2 we obtain $E_e = 0.5 * 0.1 + 0.5 * 0.5 = 0.3$

Similarly as in 1 attribute case: First, we obtain predictions of Naive Bayes, which are:

$$\begin{aligned} X_1 = T \text{ and } X_2 = T &\Rightarrow Y = T \\ X_1 = T \text{ and } X_2 = F &\Rightarrow Y = T \\ X_1 = F \text{ and } X_2 = T &\Rightarrow Y = T \\ X_1 = F \text{ and } X_2 = F &\Rightarrow Y = F \end{aligned} \tag{2}$$

Again to make calculation shorter, we only consider cases when predicted class differs from true class. We get:

$$\begin{aligned} E_e &= P(X_1 = T, X_2 = T, Y = F) + P(X_1 = T, X_2 = F, Y = F) + \\ &\quad P(X_1 = F, X_2 = T, Y = F) + P(X_1 = F, X_2 = F, Y = T) \\ &\quad \text{remember, probabilities factorize: } P(X_1, X_2, Y) = P(Y)P(X_1|Y)P(X_2|Y) \\ &= 0.5 * 0.3 * 0.1 + 0.5 * 0.3 * 0.9 + 0.5 * 0.7 * 0.1 + 0.5 * 0.5 * 0.1 = 0.235 \end{aligned} \quad (3)$$

iii. Now, suppose that we create new attribute X_3, which is an exact copy of X_2. So, for every training example, attributes X_2 and X_3 have the same value, $X_2 = X_3$. What is the error rate of Naive Bayes now?

★ **SOLUTION:** As in previous case: First, we obtain predictions of Naive Bayes. Notice that now NB also considers attribute X_3 for which it assumes that it is conditionally independent of X_1, X_2 given Y. The predictions now change, since double counting the evidence from X_2 makes the difference:

$$\begin{aligned} &X_1 = T \text{ and } X_2 = T \Rightarrow Y = T \\ &X_1 = T \text{ and } X_2 = F \Rightarrow Y = T \\ &X_1 = F \text{ and } X_2 = T \Rightarrow Y = F \quad \text{***prediction changed!} \\ &X_1 = F \text{ and } X_2 = F \Rightarrow Y = F \end{aligned} \quad (4)$$

However, the nature (the true model) remains the same. We did not introduce any new information to the system by creating X_3. So $P(X_1, X_2, X_3, Y) = 0$ if $X_2 \neq X_3$ (this never happens), and also $P(X_1, X_2, X_3, Y) = P(X_1, X_2, Y)$, since X_3 is exact (determinist) copy of X_2.
So the calculation of expected error is the same as in question (c), but now with different values of Y (remember, predictions of NB changed):

$$\begin{aligned} E_e &= P(X_1 = T, X_2 = T, Y = F) + P(X_1 = T, X_2 = F, Y = F) + \\ &\quad P(X_1 = F, X_2 = T, Y = T) + P(X_1 = F, X_2 = F, Y = T) \\ &\quad \text{probabilities then factorize as in previous case...} \\ &\quad \text{and after a bit of work...} \\ &= 0.3 \end{aligned} \quad (5)$$

■ **COMMON MISTAKE 1:** Many people calculated $P(X_1, X_2, X_3, Y)$ as $P(X_1, X_2, X_3, Y) = P(Y) * P(X_1|Y) * P(X_2|Y) * P(X_3|Y)$, but this is not true, since X_2 and X_3 are not conditionally independent given Y. Also, X_3 is a deterministic copy of X_2 so $P(X_1, X_2, X_3, Y) = 0$ if $X_2 \neq X_3$.

The nature, the true model, that generates the data did not change. By duplicating an attribute we did not introduce any new information to the system, so it makes no sense for performance to improve. Some people got the result where the error decreased, which does not make much sense, since we could get error rate going to 0 by just duplicating attribute X_2 many times.

iv. Explain what is happening with Naive Bayes?

★ **SOLUTION:** NB assumes attributes are conditionally independent given the class. This is not true, when we introduce X_3, so NB over-counts the evidence from attribute X_2 and the error increases.

v. (extra credit 10 pts) In spite of the above fact we will see that in some examples Naive Bayes doesn't do too badly. Consider the above example i.e. your features are X_1, X_2 which are truly independent given Y and a third feature $X_3 = X_2$. Suppose you are now given an example with $X_1 = T$ and $X_2 = F$. You are also given the probabilities $P(Y = T|X_1 = T) = p$ and $P(Y = T|X_2 = F) = q$, and $P(Y = T) = .5$. Prove that the decision rule is $p \geq \frac{(1-q)^2}{q^2+(1-q)^2}$ (Hint : use Bayes rule again). What is the true decision rule ? Plot the two decision boundaries (vary q between $0 - 1$) and show where Naive Bayes makes mistakes.

★ **SOLUTION: Naive Bayes The decision rule for Naive Bayes is : Predict $Y = T$ if**

$$\mathbf{Pr}\{Y = T\} \prod_i \mathbf{Pr}\{X_i = x_i \,|\, Y = T\} \geq \mathbf{Pr}\{Y = F\} \prod_i \mathbf{Pr}\{X_i = x_i \,|\, Y = F\}$$

But we know that $\mathbf{Pr}\{Y = T\} = \mathbf{Pr}\{Y = F\}$

$$\prod_i \mathbf{Pr}\{X_i = x_i \,|\, Y = T\} \geq \prod_i \mathbf{Pr}\{X_i = x_i \,|\, Y = F\}$$

Using Bayes rule again we get,

$$\prod_i \frac{\mathbf{Pr}\{Y = T|X_i = x_i\}\,\mathbf{Pr}\{Y = T\}}{\mathbf{Pr}\{X_i = x_i\}} \geq \prod_i \frac{\mathbf{Pr}\{Y = F|X_i = x_i\}\,\mathbf{Pr}\{Y = F\}}{\mathbf{Pr}\{X_i = x_i\}}$$

Now, the $\prod_i \mathbf{Pr}\{X_i = x_i\}$ parts cancel from both sides. Also $\mathbf{Pr}\{Y = T\} = \mathbf{Pr}\{Y = F\}$. Hence we now have

$$\prod_i \mathbf{Pr}\{Y = T|X_i = x_i\} \geq \prod_i \mathbf{Pr}\{Y = F|X_i = x_i\}$$

Setting $x_1 = T$ and $x_3 = x_2 = F$, we find :

$$\begin{aligned} &\mathbf{Pr}\{Y = T \,|\, X_1 = T\}\,\mathbf{Pr}\{Y = T \,|\, X_2 = F\}\,\mathbf{Pr}\{Y = T \,|\, X_3 = F\} \\ \geq\ &\mathbf{Pr}\{Y = F \,|\, X_1 = T\}\,\mathbf{Pr}\{Y = F \,|\, X_2 = F\}\,\mathbf{Pr}\{Y = F \,|\, X_3 = F\} \end{aligned}$$

Plugging in the values given in question we now obtain:

$$pq^2 \geq (1-p)(1-q)^2$$

Thus the "Naive Bayes" decision rule is given by

Predict $Y = T \iff \boxed{p \geq \frac{(1-q)^2}{p^2 + (1-q)^2}}$.

Actual Decision Rule. The actual decision rule doesn't take in consideration X_3, as we know that X_1 and X_2 are truly independent given Y and they are enough to predict Y. Therefore, we predict true if and only if $\mathbf{Pr}\{Y = T \mid X_1 = T, X_2 = F\} \geq \mathbf{Pr}\{Y = F| \mid X1_T, X2_F\}$.
Following similar calculations as before, we have

$$pq \geq (1-p)(1-q)$$
$$p \geq 1 - q$$

Thus the "real" decision rule is given by
Predict $Y = T \iff \boxed{p \geq 1 - q}$

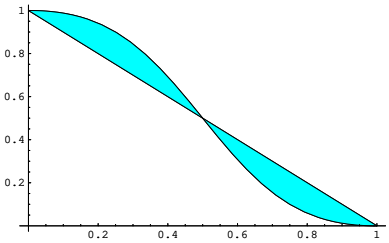

Figure 1: True and Naive-Bayes decision boundary

The curved line is for Naive bayes, and the straight line shows the true decision boundary. The shaded part in figure 1 shows the region where Naive-Bayes decision differs from the true decision.

■ **COMMON MISTAKE 2:** Some students started with $\mathbf{Pr}\{Y = T \mid X_1\}\,\mathbf{Pr}\{Y = T \mid X_2\}\,\mathbf{Pr}\{Y$ $\mathbf{Pr}\{Y = F \mid X_1\}\,\mathbf{Pr}\{Y = F \mid X_2\}\,\mathbf{Pr}\{Y = F \mid X_3\}$. Note that conditional independence does not imply this. However for this particular parameters it works out to be of this form.

2.3 Learning Curves of Naive Bayes and Logistic Regression

Compare the two approaches on the Breast Cancer dataset you can download from course webpage. You can find the description of this dataset from the course webpage. We have removed the records with missing values for you. Obtain the learning curves similar to Figure 1 in the paper.

Implement a Naive Bayes classifier and a logistic regression classifier with the assumption that each attribute value for a particular record is independently generated.

For the NB classifier, assume that $P(x_i|y)$, where x_i is a feature in the breast cancer data (that is, i is the number of column in the data file) and y is the label, of the following multinomial distribution form:

For $x_i \in \{v_1, v_2, ..., v_n\}$,

$$p(x_i = v_k|y = j) = \theta^i_{jk} \; s.t. \; \forall i, j : \sum_{k=1}^{n} \theta^i_{jk} = 1$$

where $0 \leq \theta_{jk} \leq 1$ It may be easier to think of this as a normalized histogram or as a multi-value extension of the Bernoulli.

Use 2/3 of the examples for training and the remaining 1/3 for testing. Be sure to use 2/3 of each class, not just the first 2/3 of data points.

For each algorithm:

1. (3 pts) Implement the IRLS algorithm for Logistic regression.

2. (6 pts) Plot a learning curve: the accuracy vs. the size of the training data. Generate six points on the curve, using [.01 .02 .03 .125 .625 1] fractions of your training set and testing on the full test set each time. Average your results over 5 random splits of the data into a training and test set (always keep 2/3 of the data for training and 1/3 for testing, but randomize over which points go to training set and which to testing). This averaging will make your results less dependent on the order of records in the file. Plot both the Naive Bayes and Logistic Regression, learning curves on the same plot. Use the `plot(x,y)` function in Matlab since the training data fractions are not equally spaced.

   Specify your choice of prior/regularization parameters and keep those parameters constant for these tests. A typical choice of constants would be to add 1 to each bin before normalizing (for NB) and $\lambda = 0$ (for LR).

   ★ **SOLUTION:** Please look at figure 2. Something to remember : many students treated the discrete values of each attribute categorically, i.e. have a different weight for each possible value of a feature. However note that here the values of the features mean something, i.e. strength of that feature, so its better to use them numerically. I have not penalized any of these solutions.

3. (1 pt) What conclusions can you draw from your experiments? Specifically, what can you say about speed of convergence of the classifiers? Are these consistent with the results in the NIPS paper that we have mentioned? If yes, state that. If no, explain why not.

   ★ **SOLUTION:** From figure 2 it can be observed that NB gets to its asymptotic error much faster than LR. For some of you the two curves will cross, and for some of you they will come very close but not cross. Both of these are fine because the data set is too small compared to the number of covariates to actually observe the effect reported in Ng et al's paper.

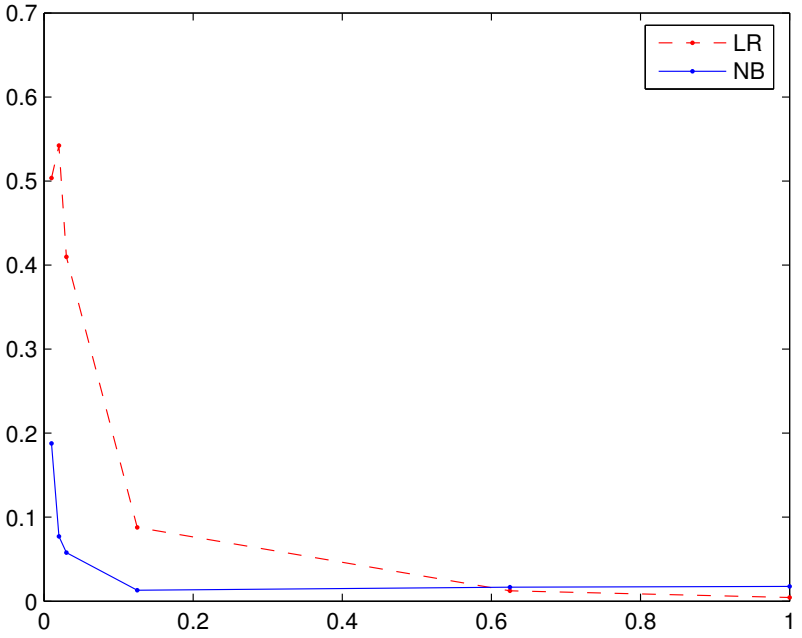

Figure 2: ★ **SOLUTION:** Your plot should look something similar to this.