10-601 Machine Learning: Homework Assignment 3

Professor Tom Mitchell
Carnegie Mellon University
February 4, 2009

- The assignment is due at 1:30pm (beginning of class) on **Wednesday, February 18, 2009**.
- Submit writeups to Problem 1 and Problem 2 *separately* with your name on each problem. Please do not staple the two writeups together.
- Write your name at the top right-hand corner of each page submitted.
- Each student must hand in their own answers to the following questions, and their own code. To submit your code, please send it as an attachment via email to `purnamritas` AT `gmail.com`. Package your code as a gzipped TAR file or a ZIP file with the prefix `601hw3-johndoe` where you substitute in your first and last names into the filename in place of '`johndoe`'. See the course webpage for the collaboration policies.
- Each question has the name of the TA who is the primary contact point for that question. Feel free to ask the other instructors about any question, but that TA is the authority on that question.

1 Naive Bayes [Andy: 35 points]

1.1 Duplicate Features and Decision Rules

The conditional independence assumptions made by Naive Bayes may not hold in reality. In spite of that fact, Naive Bayes can often do quite well at classification tasks. Consider an example where X_1, X_2 and X_3 are all Boolean features and Y is a Boolean label. X_1 and X_2 are truly independent given Y and X_3 is a copy of X_2 (meaning that X_3 and X_2 always have the same value). Suppose you are now given a test example with $X_1 = T$ and $X_2 = X_3 = F$. You are also given the probabilities:

$$P(X_1 = T|Y = T) = p$$
$$P(X_1 = T|Y = F) = 1 - p$$
$$P(X_2 = F|Y = T) = q$$
$$P(X_2 = F|Y = F) = 1 - q$$
$$P(Y = T) = P(Y = F) = 0.5$$

1. Prove that the Naive Bayes decision rule for classifying the test example positively is:

$$p \geq \frac{(1-q)^2}{q^2 + (1-q)^2}$$

2. What is the true decision rule for classifying the test example positively, in terms of p and q (Hint: it should ignore the value of X_3 in the test example)?

3. Plot the two decision boundaries (vary q between 0 and 1) and show where the first rule makes mistakes relative to the true decision rule.

1.2 Logistic Regression and Naive Bayes – Boolean Case

In section 3 of the "Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression" reading we showed that when Y is Boolean and $X = \langle X_1 \dots X_n \rangle$ is a vector of continuous variables, then the assumptions of the Gaussian Naive Bayes classifier imply that $P(Y|X)$ is given by the logistic function with appropriate parameters W. In particular:

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

and

$$P(Y = 0|X) = \frac{\exp(w_0 + \sum_{i=1}^{n} w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^{n} w_i X_i)}$$

Consider instead the case where Y is Boolean and $X = \langle X_1 \dots X_n \rangle$ is a vector of Boolean variables. Prove for this case also that $P(Y|X)$ follows this same form (and hence that Logistic Regression is also the discriminative counterpart to a Naive Bayes generative classifier over Boolean features).

Hints:

1. Simple notation will help. Since the X_i are Boolean variables, you need only one parameter to define $P(X_i|Y = y_k)$. Define $\theta_{i1} \equiv P(X_i = |Y = 1)$, in which case $P(X_i = 0|Y = 1) = (1 - \theta_{i1})$. Similarly, use θ_{i0} to denote $P(X_i = 1|Y = 0)$.

2. Notice with the above notation you can represent $P(X_i|Y = 1)$ as follows

   $$P(X_i|Y = 1) = (\theta_{i1})^{X_i}(1 - \theta_{i1})^{(1-X_i)}$$

   Note when $X_i = 1$ the second term is equal to 1 because its exponent is zero. Similarly, when $X_i = 0$ the first term is equal to 1 because its exponent is zero.

1.3 Relaxing the Conditional Independence Assumption

To capture interactions between features, the Logistic Regression model can be supplemented with extra terms. For example, a term can be added to capture a dependency between X_1 and X_2:

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + w_{1,2}X_1X_2 + \sum_{i=1}^{n} w_i X_i)}$$

Similarly, the conditional independence assumptions made by Naive Bayes can be relaxed so that X_1 and X_2 are not assumed to be conditionally independent. In this case, we can write:

$$P(Y|X) = \frac{P(Y)P(X_1, X_2|Y)\prod_{i=3}^{n} P(X_i|Y)}{P(X)}$$

Prove that for this case, that $P(Y|X)$ follows the same form as the logistic regression model supplemented with the extra term that captures the dependency between X_1 and X_2 (and hence that the supplemented Logistic Regression model is the discriminative counterpart to this generative classifier).

1. Once again, simple notation will help. To define $P(X_1, X_2|Y)$, you need more parameters than before.

   Define $\beta_{ijk} \equiv P(X_1 = i, X_2 = j|Y = k)$.

2. Notice with the above notation you can represent $P(X_1, X_2|Y = k)$ as follows

   $$P(X_1, X_2|Y = k) = (\beta_{11k})^{X_1X_2}(\beta_{10k})^{X_1(1-X_2)}(\beta_{01k})^{(1-X_1)X_2}(\beta_{00k})^{(1-X_1)(1-X_2)}$$

2 Logistic Regression (LR) and Naive Bayes (NB) [Purna: 65 points]

The data: In this assignment you will train a Naive Bayes and a Logistic Regression classifier to predict the class of a set of documents, represented by the words which appear in them. Please download the data from here[1]. The .data file is formatted "docIdx wordIdx count". Note that this only has words with nonzero counts. The .label file is simply a list of label id's. The i^{th} line of this file gives you the label of the document with docIdx i. The .map file maps from label id's to label names. In this assignment you will classify documents into two classes: rec.sport.baseball (10) and rec.sport.hockey (11). The vocabulary.txt file contains the vocabulary for the indexed data. The line number in vocabulary.txt corresponds to the index number of the word in the .data file.

2.1 Implement Logistic Regression and Naive Bayes

1. Implement regularized Logistic Regression using gradient descent. Your instructors found that learning rate η around 0.0001, and regularization parameter λ around 1 works well for this dataset. This is just a rough point to begin your experiments with, please feel free to change the values based on what results you observe. Report the values you use. One way to determine convergence might be by stopping when the maximum entry in the absolute difference between the current and the previous weight vectors falls below a certain threshold. You can use other criteria for convergence if you prefer. Please specify what you are using. In each iteration report the log-likelihood, the training-set misclassification rate and the norm of weight difference you are using for determining convergence.

2. Implement the Naive Bayes classifier for text classification using the principles in Tom's lecture in class. You can use a "hallucinated" count of 1 for the MAP estimates.

2.2 Feature Selection

1. Train your Logistic Regression algorithm on the 200 randomly selected datapoints provided in here. Now look for the indices of the words "baseball", "hockey", "nfl" and "runs". If you sort the absolute values of the weight vector obtained from LR in descending order, where do these words appear? Based on this observation, how would you select interesting features from the parameters learnt from LR?

2. Use roughly $\frac{1}{3}^{rd}$ of the data as training and $\frac{2}{3}^{rd}$ of it as test. About half the number of documents are from one class. So pick the training set with a equal number of positive and negative points (198 of each in this case). Now using your feature selection scheme from the last question, pick the [20, 50, 100, 500, all] most interesting features and plot the error-rates of Naive Bayes and Logistic Regression. Remember to average your results on 5 random training-test partitions. What general trend do you notice in your results? How does the error rate change when you do feature selection? How would you pick the number of features based on this?

2.3 Highly Dependent Features: How do NB and LR differ?

In question 1.1 you considered the impact on Naive Bayes when the conditional independence assumption is violated (by adding a duplicate copy of a feature). Also question 1.3 formulates

[1]These datasets were collected from Jason Rennie's webpage.

the discriminative analog of Naive Bayes, where we explicitly model the joint distribution of two features. In the current question, we introduce highly *dependent* features to our Baseball Vs. Hockey dataset and see the effect on the error rates of LR and NB. A simple way of doing this is by simply adding a few duplicate copies of a given feature to your dataset. First create a dataset D with the wordIds provided here. For each of the three words *baseball, hockey*, and *runs*:

1. Add 3 and 6 duplicate copies of it to the dataset D and train LR and NB again. Now report the respective average errors obtained by using 5 random train-test splits of the data (as in 2.1). For each feature report the average error-rates of LR and NB for the following:

   - Dataset with no duplicate feature added (D).
   - Dataset with 3 duplicate copies of feature added (D').
   - Dataset with 6 duplicate copies added (D").

   In order to have a fair comparison, use the same set of test-train splits for each of the above cases.

2. How do Naive Bayes and Logistic Regression behave in the presence of duplicate features?

3. Now compute the weight vectors for each of these datasets using logistic regression. Let W, W', and W'' be the weight vectors learned on the datasets D, D', and D'' respectively. You do not have to do any test-train splits. Compute these on the entire dataset. Look at the weights on the duplicate features for each case. Based on your observation can you find a relation between the weight of the duplicated feature in W', W'' and the same (not duplicated) feature in W? How would you use this observation to explain the behavior of NB and LR?