Homework #2

CS 260: Machine Learning Algorithms
Prof. Ameet Talwalkar
Due: 10/15/15, 8am
Please abide by the Academic Integrity Policy

1 Naive Bayes

The binary Naive Bayes classifier has interesting connections to the logistic regression classifier. You will show that, under certain assumptions, the Naive Bayes likelihood function is identical in form to the likelihood function for logistic regression. You will then derive the MLE parameter estimates under these assumptions.

a. Suppose $X = \{X_1, \ldots, X_D\}$ is a continuous random vector in $\mathbb{R}^D$ representing the features and Y is a binary random variable with values in $\{0, 1\}$ representing the class labels. Let the following assumptions hold:

- The label variable Y follows a Bernoulli distribution, with parameter $\pi = P(Y = 1)$.
- For each feature X_j, we have $P(X_j|Y = y_k)$ follows a Gaussian distribution of the form $\mathcal{N}(\mu_{jk}, \sigma_j)$.

Using the Naive Bayes assumption that states "*for all $j' \neq j$, X_j and $X_{j'}$ are conditionally independent given Y*", compute $P(Y = 1|X)$ and show that it can be written in the following form:

$$P(Y = 1|X) = \frac{1}{1 + \exp(-w_0 + \mathbf{w}^\top \mathbf{X})}.$$

Specifically, you need to find the explicit form of w_0 and $\mathbf{w}$ in terms of π, μ_{jk}, and σ_j, for $j = 1, \ldots, D$ and $k \in \{0, 1\}$.

b. Suppose a training set with N examples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$ is given, where $\mathbf{x}_i = (x_{i1}, \cdots, x_{iD})^\top$ is a D-dimensional feature vector, and $y_i \in \{0, 1\}$ is its corresponding label. Using the assumptions in 1.a (not the result), provide the maximum likelihood estimation for the parameters of the Naive Bayes with Gaussian assumption. In other words, you need to provide the estimates for π, μ_{jk}, and σ_j, for $j = 1, \ldots, D$ and $k \in \{0, 1\}$.

2 Logistic Regression

Consider a binary logistic regression model, where the training samples are *linearly separable*.

a. Given n training examples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_n, y_n)$ where $y_i \in \{0, 1\}$, write down the negative log likelihood, $\mathcal{L}(\mathbf{w})$, in terms of the sigmoid function, x, and y.

b. Is this loss function convex? Provide your reasoning.

c. Show that the magnitude of the optimal $\mathbf{w}$ can go to infinity when the training samples are *linearly separable*.

d. A convenient way to prevent numerical instability issues is to add a penalty term to the likelihood function as follows:

$$\mathcal{L}(\mathbf{w}) = -\log\left(\prod_{i=1}^{n} p(Y = y_i|X = x)\right) + \lambda\|\mathbf{w}\|_2^2, \tag{1}$$

where $\|\mathbf{w}\|_2^2 = \sum_i w_i^2$ and $\lambda > 0$. Compute the gradient respect to w_i, i.e. $\frac{\partial \mathcal{L}(\mathbf{w})}{\partial w_i}$.

e. Show that the problem in Eq. (1) has a unique solution.

3 Decision Trees

a. Suppose you want to grow a decision tree to predict the *accident rate* based on the following accident data which provides the rate of accidents in 100 observations. Which predictor variable (weather or traffic) will you choose to split in the first step to maximize the information gain?

Weather	Traffic	Accident Rate	Number of observations
Sunny	Heavy	High	23
Sunny	Light	Low	5
Rainy	Heavy	High	50
Rainy	Light	Low	22

b. Suppose in another dataset, two students experiment with decision trees. The first student runs the decision tree learning algorithm on the raw data and obtains a tree T_1. The second student, normalizes the data by subtracting the mean and dividing by the variance of the features. Then, he runs the same decision tree algorithm with the same parameters and obtains a tree T_2. How are the trees T_1 and T_2 related?

c. Choosing a splitting criterion is a crucial component of a decision tree algorithm. As discussed in ESL, Section 9.2.3, the most common splitting criteria are the *Gini index* and *Cross-entropy*. Both of these can be viewed as convex surrogates for the misclassification error. Prove that, for any discrete probability distribution p with K classes, the value of the Gini index is less than or equal to the corresponding value of the cross-entropy. This implies that the Gini index more closely approximates the misclassification error.

Definitions: For a K-valued discrete random variable with probability mass function $p_i, i = 1, \ldots, K$ the Gini index is defined as: $\sum_{k=1}^{K} p_k(1 - p_k)$ and the cross-entropy is defined as $-\sum_{k=1}^{K} p_k \log p_k$.

4 Comparing Classifiers in MATLAB/Octave

In this problem, you will work with the same dataset as in HW1, and compare the performance of various classification algorithms. Starting with the one-hot-encoded version of the data that you generated in Question 4a in HW1, perform the following steps:

a. Fill in the function `naive_bayes` in the `naive_bayes.m` file. In particular, implement the Bernoulli Naive Bayes model from scratch (this will first require you to compute the MLE estimates). The inputs of this function are training data and new data (either validation or testing data). The function needs to output the accuracy on both training and new data (either validation or testing). Note that some feature values might exist in the validation/testing data that do not exist in the training data. In that case, please set the probability of that feature value to a small value, for example, 0.1. Note: You should NOT use any related Matlab toolbox functions, e.g., `NaiveBayes.fit` to implement Naive Bayes.

b. Compare the four algorithms (kNN, Naive Bayes, Decision Tree, and Logistic Regression) on the provided dataset. For each algorithm, report accuracies as detailed below, and describe the relative performance of these algorithms in a few sentences.

***k*NN:** Report results from HW1.

Decision Tree: Train decision trees using the function `ClassificationTree.fit` or `fitctree` in Matlab. Report the training, validation and test accuracy for different split criterions (*Gini index* and *cross-entropy* using the SplitCriterion attribute) and different settings for the minimum size of leaf nodes to $1, 2, \cdots, 10$ (using the MinLeaf attribute). Thus, in total you will report the results for $2 \times 10 = 20$ different cases. When training decision trees, turn off pruning using the `Prune` attribute.

Naive Bayes: Report the training, validation and test accuracy.

Logistic Regression: Train multi-class logistic regression using the function `mnrfit` in Matlab. Report the training, validation and test accuracy.

Submission

Please provide the following as part of your submission:

- Provide your answers to problems 1-3 and 4b in hardcopy. The papers need to be stapled and submitted at the beginning of class on the due date.
- Email all source code to Nikos (nikos.karianakis at gmail) by the due date. The only acceptable languages are MATLAB and Octave.
- List all of your collaborators on this HW.