CS57300 Homework — Homework 1 , Problem 1 — (March 11, 2016)
Jialiang Yu (0028161618)

Q1:

Nearly 2,000 users are asking for personal loans at a bank (test data). You need to decide if the bank should or should not lend money to these costumers. The bank should not lend money for costumers that do not repay the full amount they borrowed. Download the training, test, and submission example data $https : //www.cs.purdue.edu/homes/ribeirob/courses/Spring2016/hw/midterm/bank_data.zip$.

(a) Describe how we can use a Naive Bayes Classifier (NBC) classifier for this problem.

  (i) (Theory) Describe the equations, the inputs, and how we can classify out-of-sample items with the output.

  (ii) (Theory) Describe the importance of priors in NBC.

(b) Describe how we can use a logistic regression classifier for this problem.

  (i) (Theory) Describe the equations, the inputs, and how we can classify out-of-sample items with the output.

  (ii) What happens if $\phi^T R \phi$ (slide 34 of Lecture 14) is singular? What is the reason and how can you solve this issue?

(c) Describe how we can use a SVM classifier for this problem.

  (i) (Theory) Describe the equations, the inputs, and how we can classify out-of-sample items with the output.

  (ii) (Theory) What is the advantage of a SVMs over linear regression for classification?

  (iii) (Empirical) Which kernel, linear or Gaussian seems to work best with the data? Can you give a short likely explanation of why?

(d) (Empirical) Use K-fold validation for $K \in \{5, 10\}$ to estimate the average F1 score of each of the above classifiers (NBC, Logistic, SVM including linear and Gaussian kernels), where N is the number of observations. Remember to describe K-fold validation and how you obtained the average F1 score (the average F1 score is just the average of the obtained K F1 scores). For the Logistic Regression classifier plot the ROC curve and give the Area Under the Curve (AUC) score.

(e) (Empirical) Using $50 - fold$ validation perform a paired t-test to decide which classifier has the best F1 score. Is the F1 score of the best classifier significantly different ($at \alpha = 0 : 05 level$) than the other three classifiers?

(f) (Empirical) Choose what you think is your best solution .

A:

(a) (i) Naive Bayes classifiers can handle an arbitrary number of independent variables whether continuous or categorical. Given a set of variables, $X = x_1, x_2, ..., x_d$, we want to construct the posterior probability for the event C_j among a set of possible outcomes $C = c_1, c_2, .._{,d}$. Using Bayes' rule:

$$p(C_j|x_1, x_2, ..., x_d) \propto (x_1, x_2, ..., x_d|C_j)p(C_j)$$

where $p(C_j|x_1, x_2, ..., x_d)$ is the posterior probability of class membership, i.e., the probability that X belongs to C_j. Since Naive Bayes assumes that the conditional probabilities of the independent variables are statistically independent we can decompose the likelihood to a product of terms:

$$p(X|C_j) \propto \prod_{k=1}^{d} p(x_k|C_j)$$

and rewrite the posterior as:

$$p(C_j|X) \propto p(C_j) \prod_{k=1}^{d} p(x_k|C_j)$$

Using Bayes' rule above, we label a new case X with a class level C_j that achieves the highest posterior probability.

(ii) Although the assumption that the predictor variables are independent is not always accurate, it does simplify the classification task dramatically, since it allows the class conditional densities $p(x_k|C_j)$ to be calculated separately for each variable, i.e., it reduces a multidimensional task to a number of one-dimensional ones. In effect, Naive Bayes reduces a high-dimensional density estimation task to a one-dimensional kernel density estimation. Furthermore, the assumption does not seem to greatly affect the posterior probabilities, especially in regions near decision boundaries, thus, leaving the classification task unaffected.

(b) (i) For a linear regression classifier, the input is the training set which is described as a matrix with each row representing a sample point. Let m be the number of examples and each data point has features as $x = [x_0, x_1, x_2, ..., x_n]$. Let $h_\theta(x) = \frac{1}{1-e^{-\theta^T x}}$ be the estimate function of a given sample x which estimates the probability that $y = 1$ for a given x.

Let

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

and try to minimize $J(\theta)$.

For $j = 0, 1, 2, ..., m$, repeat updating θ_j with

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) = \theta_j - \alpha [\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}]$$

The classification of out-of-sample items is given by $\widehat{y} = \theta^T \widehat{x}$.

(ii) If $\phi^T R \phi$ is singular, then $(\phi^T R \phi)^{-1}$does not exist.

(c) (i) For linear model with non-linear features ϕ

$$y(x) = w^T \phi(x) + b$$

For two classes, if class $t_n \in \{-1, 1\}$ of item x_n, then $t_n y(x_n) > 0$ means correctly classified. Thus, distance of x_n from decision hyperplane is the maximum minimum distance

$$\frac{t_n y(x_n)}{||x||} = \frac{t_n(w^T \phi(x_n) + b)}{||w||}$$

We need to find

$$\underset{w,b}{argmax} \left\{ \frac{1}{||w||} \min_n [t_n(w^T \phi(x_n) + b)] \right\}$$

To modify the solution, we can calculate

$$\underset{w,b}{argmin} \frac{1}{2} ||w||^2$$

s.t.

$$t_n(x^T\phi(x_n)+b)\geq 1,\ n=1,...,N$$

(ii) When two classes of , say points mix together on the graph, the SVM could use its kernel to find a proper hyperplane to separate the two classes while the linear regression only has the straight line coarsely separate the two classes. Besides, when the points from the same class group into several sets, the linear classifier is hard to find the board between two classes properly, however, the SVM could ignore the sets far away from the board and provides more accurate classification.

(iii) The Gaussian kernel gives the better result. The linear kernel gives 62.8% accuracy while the Gaussian kernel gives 63.2% accuracy. The linear kernel is a degenerate version of RBF, bence the linear kernel is never more accurate than a properly tuned RBF kernel.

(d) The K-fold validation is a method that split the data set into K folds and use one of the K folds as a test set, the other ones as training sets. Repeat the process until each fold has been treated as a test set once. F1 score, also known as F-score or F-measure can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. The formula of F1 score is :

$$F_1 = 2*\frac{precision*recall}{precision+recall}$$

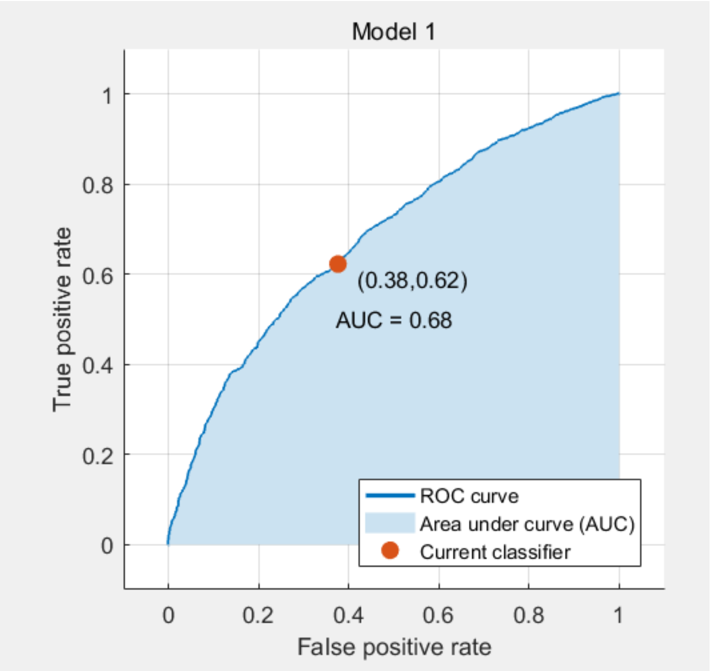
The average F1 score is obtained by

$$\frac{1}{K}\sum_{i=1}^{K}F_i$$

When $K=5$, the F1 score of {NBC, Logistic, SVM-linear kernel, SVM-Gaussian kernel} is {0.575, 0.62278, 0.628, 0.632}.

When $K=10$, the F1 score of {NBC, Logistic, SVM-linear kernel, SVM-Gaussian kernel} is {0.575, 0.6225, 0.631, 0.631}.

For the Logistic Regression classifier, the ROC curve and the Area Under the Curve (AUC) score are shown below, the ROC curve is plotted in Matlab:



(e) Let $X_1, ..., X_k \sim N(\mu, \sigma^2)$ where both μ and σ^2 are unknown and k is relatively small. We want to test $H_0 : \mu = \mu_0$. Let the accuracy in each fold be $a_{A1}, ..., a_{Ak}$ for algorithm A, and $a_B1, ..., a_Bk$ for algorithm B. We assume that the pairwise differences $x_i = a_{Ai} - a_{Bi}, i = 1...k$ follow $N(0, \sigma^2)$ under H_0. Therefore T has a t-distribution with k-1 degree of freedom. We can look up the 0.05 threshold $(2 - sided)$ from a table. When T is outside the threshold we reject H_0, and claim that A and B are truly different.

For $K = 50$, the p-value paired t-test of NBC and Logistic is $0.00052 < 0.05$, so we accept the H_0 that Naive Bayes Classifier and Logistic Regression have the same effect. The p-value of classifier pairs are like below :

$(NBC, Logistic) = 0.000520$, $(NBC, SVM_{linear}) = 0.00297$, $(NBC, SVM_{Gaussian}) = 1.0e^{-34}$, $(logistic, SVM_{linear}) = 4.47e^{-6}$, $(logistic, SVM_{Gaussian}) = 2.93e^{-37}$, $(SVM_{linear}, SVM_{Gaussian}) = 0.000869$.

It seems that there is no difference between those classifiers, but I think the $SVM_{Gaussian}$ classifier is the best.

(f) I choose SVM with Gaussian kernel as the best solution. The predict result is shown in the Submission.csv file.

Q2:

A:

(a) (i) Shortest Path: Proximity measured by length of shortest path between u and v. Suggests connections between nodes that are nearby.

Problem:Network diameter often very small and distribution very concentrated.

(ii) Common Neighbours: Common neighbours uses as score the number of common neighbours between vertices u and v.

$$score(u, v) = |N(u) \cap N(v)|$$

Problem: Large scores for vertices with too many neighbours.

(iii) Jaccard Similarity:

$$score(u, v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

It fixes the Common Neighbour's problem that u and v have too many neighbours problem by dividing the intersection by the union. But vertices with too many neighbours will get very low score.

(iv) Adamic-Adar: This score gives more weitght to neighbours that are not shared with many others.

$$score(u, v) = \sum_{z \in N(u) \cap N(v)} \frac{1}{log|N(z)|}$$

(v) Katz score: Build binary symmetric adjacency matrix A based on the connection condition and calculate

$$score(u, v) = \sum_{l=1}^{\infty} \alpha^l (A^l)_{u,v}$$

It is the exponentially weighted sum of number of paths of length l

(vi) PageRank: Stationary probability walker is at v under the following random walk: with probability α, jump back to u, with probability $1 - \alpha$, go to random neighbour of current node.

$$score(u, v) = \pi_v^{(u)}$$

(b) With this problem, I choose Jaccard Similarity as the solution. The result is shown in the result.csv file. (The program runs too long which is expected, I have run the program since 5:00 p.m. today and until now it's still running. So I just submit the code with the prediction file empty. The details are shown in the code.)