An Empirical Comparison of Five Supervised Learning Algorithms – KNN, SVMs, DT, Bagged DT and Naive Bayes

Xuanpei Ouyang
Professor Zhuowen Tu
COGS 118A
Final Project

***Abstract*—This report examines and compares the performance of five supervised machine-learning algorithms on four different binary classification tasks, including balanced and unbalanced classification tasks. The machine learning algorithms included in this paper are K Nearest Neighbor, Support Vector Machines, Naïve Bayes, Decision Tree and Bagged Decision Tree. The methods in this report mostly follows the procedure used in *An Empirical Comparison of Supervised Learning Algorithms by* Rich Caruana and Alexandru Niculescu-Mizil (See Caruana & Niculescu-Mizil, 2006). This report mostly focuses on comparing the performance of different classifiers on different tasks by analyzing the Accuracy, ROC curve, F-score and squared error.**

***Index Terms*—Bootstrap Analysis, Decision Tree, K Nearest Neighbor, Random Forests, Supervised Machine Learning Algorithms, Support Vector Machine, Weighted K Nearest Neighbor, ACC, F-score, LFT, ROC, Average Precision, Precision/Recall break even point, Squared error, Cross-entropy, UCI Repository**

I. Introduction

In supervised machine learning, the model is developed by a labeled train data set consists of pairs of input and output. All the data are labeled which means each data include an input and a correct output/label. By studying the relationship between input and correct output, supervised learning will build a model that can predict the output value given input value. All the algorithms have their own advantages as well as limitations. I will briefly introduce the methods and algorithms used in this experiment and detailed explanation will be provided in the Methodology section. I evaluate the performance of SVMs, naive bayes, decision tree, KNN and bagged decision tree on four binary classification tasks with four performance metrics: accuracy (ACC), F-score (FSC), ROC curve and squared error (RMS). Support Vector Machines (SVMs) is one of the most popular supervised machine-learning algorithms used for classification. SVMs training algorithm is a non-probabilitic machine-learning algorithm which learn to build its model by classifying points in the feature space. In this report, in additional to use SVMs to perform linear classification, I also perform non-linear classification (kernel trick) to for better testing result. K Nearest Neighbor (KNN), a non-parametric method will choose the majority vote among the closest k neighbors. However, it only approximately based on the local information instead of overall data structure. This algorithm is very simple and does not require explicit training, while the choice of K and the formula/weights used for computing distance is very important. In this report, I test KNN algorithm with normal Euclidean distance, weighted Euclidean distance with more emphasis on closest neighbors and locally weighted average with different kernel width. Naive Bayes classifier is also a rather simple machine learning algorithm which based on Bayes' theorem and the assumption about the independence between features but still sometimes gives accurate prediction like other complicated algorithms do. Naive Bayes classifier simplifies learning by assuming that features are independent given each class and I use three types of kernel smoothing density estimate along with Naive Bayes classifier. Decision Tree, a non-parametric supervised learning algorithm, build a tree-structure model to predict label for binary classification. Decision Tree may have seriously overfitting problem because decision tree can grow to very deep and complicated to accommodate all the training data but testing data. Here, I control the maximum number of leaves and prune the decision tree to avoid overfitting. The decision tree does not work well, creating a biased decision tree, in a biased data set, such as LETTER.p1. Also, because the instability of decision tree classifier, small change in the training data set will end up with a completely different decision tree, I use bootstrap aggregating (Bagged Decision Tree) decision tree, a ensemble meta-algorithm, to lower the variance, improve the stability and accuracy of decision tree algorithm. In methodology section, I will also discuss the datasets and preprocessing of datasets before experiment. Then I will report and analyze the training and testing result and performance by metrics and by problems respectively in section 3 and 4.

II. METHODOLOGY

This section will include some details about supervised learning algorithms and information as well as the usage about three data sets from UCI repository. For most of the classifier, I tried to follow the settings provided in Rich Caruana and Alexandru Niculescu-Mizil's paper.

A. Learning Algorithm

Support Vector Machine (SVMs): I use C.C. Chang and C.J. Lin's SVM implementation - LIBSVM to train SVM models with linear kernel function, polynomial kernel function with degree 2 & 3, radial basis function with width {0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2}. For each SVM models with different kernel function, I also vary the regularization parameter C from 10^{-7} to 10^{3} for each kernel function for cross validation.

K Nearest Neighbors (KNN): For KNN algorithm, I use the Machine Learning Toolbox library provide by MATLAB to train the data sets. I use 26 values of K from $K = 1$ to $K = |trainset|$ as reported in the original paper. Also I use KNN with Euclidean distance, weighted Euclidean distance and locally weighted averaging between 2^0 and 2^{10} by a factor of 10. For the weighted Euclidean distance kernel function and the locally weighted averaging, I use the $weight_k = \frac{1}{distance}$ and $weight_k = \frac{1}{e^{Kernel\,Width * distance}}$ respectively.

Decision Tree (DT): For Decision Tree Algorithm, I use the Decision Tree implementation in Machine Learning Toolbox library provided by MATLAB to train the data sets. To avoid overfitting, I use cross validation to choose the best leafs size to indirectly control the maximum depth of decision tree model and then prune the tree to the best level to get the Decision Tree model.

Bootstrap Aggregating Tree (BAG-DT): For Bagged Decision Tree Algorithm, I followed the paper and bag 100 decision trees and find the best minimum leafs size for 100-bagged tree. Then I use cross validation to find the best level to prune the bagged tree.

Naive Bayes (NB): For Naïve Bayes Algorithm, I also use the implementation in Machine Learning Toolbox library provided by MATLAB. I train and compare three NB models with Gaussian kernel, uniform kernel and Epanechnikov kernel and get the average performance of Naïve Bayes algorithm and the performance associated with the best kernel function.

B. Performance Metrics

To accessing the performance of each classifier, I use four threshold metrics, which are accuracy (ACC), F-score (FSC), true-positive rate (TPR) and G-measure for all five learning algorithms.

C. Data Sets

I use the three data sets: ADULT, COV_TYPE and LETTER from UCI repository. I convert the COV_TYPE dataset to a binary classification problem by setting the class with largest size as positive and all the others are negative. For LETTER dataset, I followed in Rich Caruana and Alexandru Niculescu-Mizil's paper to convert LETTER dataset to two different binary classifications, one is unbalanced with negative dominating and the other is balanced. LETTER.p1 will set the letter "O" to positive and all the rest to negative. LETTER.p2 will set the letter from "A" to "M" to positive and "N" to "Z" to negative. And for ADULT dataset, I convert all the features to numbers from 1 to |*feature*| and then normalize all the features. For all four binary classification tasks, I normalize the numerical input data to rescale them into the range {0 - 1} and transform the categorical input data into 1's and 0's by one hot encoding. Since ADULT and COV_TYPE are too large, I perform a PCA to reduce the dimensionality before training and testing. Also in ADULT dataset, I replace the missing value with the median value of that feature contains the missing value.

III. EXPERIMENT

During training process, I randomly select 5000 data samples for training dataset and the rest of the data samples for testing. Similar to procedure in Rich Caruana and Alexandru Niculescu-Mizil's paper, I use 5-fold cross validation on the 5000 training data samples and the 4000 data samples are used for training classifier and the 1000 data samples are used for finding the best parameter/settings.

A. Performance by Metrics

Table 1 shows the average performance of the average performance of all the classifiers on these four binary classification tasks. In the MODEL column, the best classifiers without best settings from the same type of classifiers are **boldfaced**. The metrics I used are all threshold metrics. For calculating these performance measures, I used the MATLAB function provided by Bernan D. on MATLAB website and collect the result for accuracy, precision, recall (TPR), F-Measure and G-Measure. Overall, SVM with radial basis kernel function and bagged decision tree have the best accuracy and F-score while the Naive Bayes classifier has the lowest overall accuracy. Since F-score and G-measure are all based on Precision and Recall, I will consider them as one metric later while analyzing the data in the table.

The Accuracy (ACC) in the first column shows the overall probability that the classifier correctly predict the label for testing data and bagged decision tree algorithm has the highest accuracy rate. We can see that among SVMs modeled by different kernel functions, the accuracy of linear and degree-2 polynomial kernel function is less than 90% and the degree-3 polynomial kernel function and radial basis kernel function

TABLE 1 Average Score for Each Supervised Learning Algorithm By Metrics

MODEL	*ACC*	*FSC*	*Precision*	*Recall*	*G-measure*
SVM-Linear	0.8447	0.6850	0.7169	0.6753	0.7168
SVM-Poly2	0.8872	0.7570	0.8048	0.7223	0.8073
SVM-Poly3	0.9393	0.8679	0.8635	0.8727	0.9020
SVM-RBF	0.9535	0.8895	0.8766	0.9027	0.9208
DT	0.9369	0.7378	0.8252	0.6728	0.8009
BAG-DT	0.9403	0.7843	0.8233	0.7526	0.8494
KNN-GAU	0.8772	0.7068	0.7758	0.6726	0.7943
KNN-WGAU	0.8772	0.7110	0.7714	0.6814	0.7993
KNN-AVG	0.9276	0.8880	0.8605	0.8535	0.9007
NB-GAU	0.8528	0.3583	0.5368	0.4172	0.5150
NB-UNIFORM	0.8320	0.3342	0.5192	0.4385	0.5041
NB-EPANE	0.8435	0.3565	0.5204	0.4280	0.5387

produce better predictions. Also as expected, Naive Bayes classifier with either kernel distribution cannot give good predictions since it based on the impractical assumption that all the features are independent.

Since accuracy may sometimes leads to biased conclusion on some dataset and we also need to look at the F-score and TPR to compare the performance of each classifier. The F-score, by equally measuring precision and recall, show how effective each classifier is. We can see that Naive Bayes does not have an efficient learning algorithm the precision and recall for Naive Bayes classifiers is only around 0.5 while the accuracy for Naive Bayes can be as high as 0.8.

B. Performance by Problem

Table 2 shows the accuracy of all five classifier with best settings on the four binary classification tasks. In the table, the classifier with best performance on ACC is **boldfaced** and the classifier with second best performance is marked with *. Note that the LETTER.p1 is a biased dataset with dominating negative label data and the rest of datasets are all unbiased dataset. Overall, SVM and Bagged decision tree give the best and effective prediction among all five supervised machine learning algorithms.

Also since Naive Bayes make prediction based on the assumption of independence, we can see that the accuracy result for ADULT dataset is much lower than the accuracy result for COV_TYPE and LETTER.p2, whose features do not imply much dependence. Naive Bayes does not work well on LETTER.p1 since the process of biased converting the 26 categories of LETTER.p1 dataset greatly weakens the assumption of independence. While the performance of SVMs, KNN, DT and Bagged DT are consistent on all four binary classification tasks.

TABLE 2 ACC Score for Each Best Supervised Learning Algorithm by Problems

Model	*ADULT*	*COV_TYPE*	*LETTER.p1*	*LETTER.p2*
SVM	**0.8723**	**0.9703**	0.9180	**0.9889**
NB	0.7842	0.9047	0.7621	0.9611
KNN	0.8232	0.9547	0.9345*	0. 9894
DT	0.8687	0.9655	0.9306	0.9829
BAG-DT	0.8691*	0.9697*	**0. 9427**	0.9879*

C. Performance by Speed

In term of speed, SVMs classifiers take the longest time to run to get the output and KNN classifier is the second slowest algorithm. One reason is that the cross validation range for SVMs and KNN are large. For SVMs, the models with linear kernel function and degree-2 polynomial kernel function converge faster than the models with degree-3 or higher polynomial kernel function and radial basis kernel function.

On the other hand, decision tree and Naive Bayes algorithms are the two fastest algorithms to build a

classification models. The bagged tree algorithm takes slightly longer time to produce the result model than the decision tree algorithm does since it aggregating 100 decision trees and make prediction based on all 100 created decision trees in this case.

IV. Conclusions

In summary, SVMs and bagged decision tree work consistently well on all four dataset and Naive Bayes is the only classifier that cannot give good performance on three of the problems but COV_TYPE. We should also notice that there is not universal best classifier for all the problems and the performance given by different classifiers are greatly affected by the choice of kernel function and setting parameters. In order to maximize the performance of classifier, we should carefully choose appropriate learning algorithm and find best kernel function, choice of parameter and other settings. Also, when evaluating the performance of classifier, we should use multiple metrics since single measurement usually cannot provide comprehensive information and might leads to biased conclusion. And sometimes, there is a tradeoff between the running time and accuracy and efficiency performance of an algorithm. There are a great number of supervised machine-learning algorithms that are expected to have excellent performance such as boosted tree, random forest and neural network in additional to the five learning algorithms I reported here.

References

Caruana. R and Niculescu-Mizil. A, “An Empirical Comparison of Supervised Learning Algorithms,” Dept. of Computer Science, Cornell Univ. Ithaca, NY 2016.

Caruana. R, Karampatziakis. N and Yessenalina. A, “An Empirical Evaluation of Supervised Learning in High Dimensions,” Dept of Computer Science, Cornell Univ. Ithaca, NY 2016.

Freund. Y and Schapire. R, “A Short Introduction to Boosting,” AT&T Lab., Shannon Lab., Avenue Florham Park, NJ, 1999.

Peterson, Leif E. "K-nearest neighbor." *Scholarpedia* 4.2, 2009.

Dudani, Sahibsingh A. "The distance-weighted k-nearest-neighbor rule."*Systems, Man and Cybernetics, IEEE Transactions*, 1976.

Rish, Irina. "An empirical study of the naive Bayes classifier." *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. No. 22. IBM New York, 2001.

Weiss, Sholom M., and Ioannis Kapouleas. "An empirical comparison of pattern recognition, neural nets and machine learning classification methods." *Readings in machine learning*, 1990.