Machine Learning Algorithm for Classification

Haoyuan Tan1*

1College of Letter and Science, University of California, Davis, One Shields Avenue, Davis, CA, 95616, USA
*thytan@ucdavis.edu

Abstract. Recently, machine learning methods have a good performance in the field of classification tasks. Summarizing and comparing the performances of different classifiers in the application of their specific classification tasks has a reference significance. In this paper, five classical machine learning classifiers, including GMM, Random Forest, SVM, XGBoost, and Naive Bayes, are compared to show their computing characteristics. The advantages and disadvantages are analysed in this paper. Based on the different datasets, namely different specific classification tasks, the different classifiers perform similarly. However, the SVM-based classifier has the lowest accuracy while processing the text data to apply the text classification task. This result shows that if the classification task is difficult, the accuracy would not be high. This research summarizes the performances of different machine learning methods in the application of specific classification tasks. And this research has a reference significance for the machine learning-based classifiers.

1. Introduction

In the field of data mining, many mistakes would be made throughout the analyses or attempting to establish relationships between multiple features. The problems are challenging to solve. The Machine Learning methods are a powerful tool in applying data mining, which can effectively solve the above problems [1], advancing the efficiency of machines and the designs.

The same set of features should be used to represent the instances in any dataset by machine learning. The features could be binary, categorical, or continuous. Supervised learning is when instances are provided with known labels, which are the corresponding correct outputs. On the contrary, unsupervised learning is when instances are unlabeled. In this paper, the supervised learning task would be discussed to compare the performances of the machine learning-based classifiers.

Supervised classification is most frequently implemented by what is called the Intelligent Systems. Also, based on the Perceptron-based techniques [7, 8], the Statistics (Bayesian Networks, Instance-based techniques) [9, 10], and Logical/Symbolic techniques [2-6], a vast number of techniques have been suggested. Especially, the statistical methods and perception-based methods are prevalent in the application of classification tasks. For example, L. Torlay et al. [11] applied a statistical approach to identify atypical language modes and distinguish patients with epilepsy from healthy subjects, based on their cerebral activity, as evaluated by functional MRI (fMRI). Yonghong Huang et al. [12] introduced and evaluated the use of Gaussian mixture models (GMMs) for multiple limb motion classification using continuous myoelectric signals. The critical point of their work is to optimize the configuration of this classification scheme. In addition, in order to learn from a group of training instances, the perceptron algorithm needs to run repeatedly through the entire training set until it finds a prediction vector, which would be accurate on all of the training sets. Then use the prediction rule to predict the labels on the test
set. The most famous representative of statistical learning algorithms is the Bayesian networks, which are formed of directed acyclic graphs containing the unobserved node and observed nodes, with a strong assumption of independence among observed nodes in the context of the unobserved nodes.

Machine Learning techniques in the field of classification tasks are advantageous. Several Machine Learning application-oriented articles can be found in the previous researches [13]. For example, we all know that when dealing with multidimensions and continuous features, artificial neural networks and SVMs tend to perform more reliable [1]. Additionally, when dealing with discrete or categorical features, logic-based systems tend to perform better. In order to achieve maximum prediction accuracy, a large sample size is required for neural network models and SVMs; nevertheless, Naive Bayesian networks (NB) may need a comparatively small dataset [1]. SVMs are binary algorithms; therefore, error-correcting output coding (ECOC) should be used. Namely, the output coding approach can diminish a multi-class issue to a set of multiple binary classification problems [1]. Moreover, most decision tree algorithms cannot perform appropriately when there are obstacles that require diagonal partitioning. Accordingly, the resulting regions after partitioning are all hyperrectangles. When multicollinearity is present, both ANNs and SVMs operate well, and a nonlinear correlation exists between the input and output features [1].

According to the above description of classification tasks and Machine Learning techniques, it is easily found that the different Machine Learning methods have different advantages and disadvantages, which depend on their own characteristics of computing methods. In addition, the different classification tasks also have different characteristics because the classification tasks have their data features. Therefore, analysis of different Machine Learning methods in different classification tasks has great significance.

The rest parts of this paper include: Section 2 introduces the different Machine Learning methods in classification tasks; Section 3 presents the comparison results of different Machine Learning methods during the process of classification dataset; Section 4 gives the conclusion of the analysis of the different Machine Learning methods in classification tasks.

2. Comparison of Machine Learning Methods in Classification Tasks
The classical Machine Learning methods, such as GMM, SVM, Random Forest, XGBoost, Naive Bayes, ANN, and the Decision Tree, have been successfully applied into the field of many classification tasks. In this part, the comparison of the above classical Machine Learning methods are given to show the characteristics of their computing.

2.1. GMM Classifier
The Gaussian Mixture Model (GMM) is described by Douglas Reynolds [14], a weighted sum of Gaussian component densities is represented by a parametric probability density function. GMMs are usually used as a parametric model of the probability distribution of continuous measurements or features in a biometric system [15-17]. GMM usually has the best performance for identification and verification systems. GMM is very flexible for cluster covariance [14].

Generally, the probability density function can be described as Formula (1),

\[
p(\mathbf{x}|\lambda_n) = \sum_{i=1}^{M} w_i^n p_i^n(\mathbf{x})
\]

Where M is the number of mixture components; \( w_i^n \) represents the mixture weights and that satisfy \( \sum_{i=1}^{M} w_i^n = 1 \).

2.2. SVM Classifier
SVM proposed by Burges is developed initially for classification, and it also has been used for regression and preference (or rank) learning. The original form of SVMs is a binary classifier where the output of the learned function is either positive or negative. In order to combine multiple binary classifiers, the multiclass classification can be performed by using the pairwise coupling method [18-23]. SVM Classifier usually transforms the data and finds the optimal boundary between the outputs.

The training problem in SVM becomes a constrained optimization problem as Formular (2)
\[ \text{minimize:} \quad Q(w) = \frac{1}{2}||w||^2 \]

\[ \text{subject to:} \quad y_i(w \cdot x_i - b) \geq 1, \forall (x_i, y_i) \in D \quad (2) \]

Especially, the factor of 1/2 is used for mathematical convenience.

2.3. Random Forest Classifier

The random forest classifier consists of a combination of tree classifiers that each tree classifier generates using a random vector sampled individually from the input vector, which in combination would make up the random forest classifier. Each tree calculates a unit vote for the most popular class to classify an input vector [24].

The Gini Index, an attribute selection measure that measures the impurity of an attribute according to the classes is adapted by the random forest classifier. For a given training set T, select one case (pixel) at random and say it applies to some class \(C_i\), the Gini index can be written as formula (3)

\[ \sum \sum_{j \neq i} \left( \frac{f(C_i|T)}{|T|} \right) \left( \frac{f(C_j|T)}{|T|} \right) \quad (3) \]

where \(f(C_i, T)/|T|\) is the probability that the selected case applies to the class \(C_i\).

2.4. XGBoost Classifier

The Extreme Gradient Boosting (XGBoost) algorithm proposed by Pedregosa F et al. [25] is usually used for classification. The details can be found in Formula (4)

\[ y_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = y_i^{(t-1)} + f_t(x_i) \quad (4) \]

XGBoost Classifier is different from other algorithms, and it is an open-source library or a framework for algorithms. It is more likely to be a boosting framework for algorithms.

2.5. Naive Bayes Classifier

Naive Bayesian networks (NB) are elementary Bayesian networks formed of directed acyclic graphs containing the unobserved node and observed nodes, with a strong assumption of independence among observed nodes in the context of the unobserved nodes [26]. Therefore, the independence model (Naive Bayes) is based on estimating as Formula (5)

\[ P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (5) \]

A represents the first event, B is the second. P(A|B) means when B event is given, what is the probability that A event would happen.

Naive Bayes Classifier is a simple method for creating classifiers. Using the method of maximum likelihood, Naive Bayes classifiers can be trained efficiently.

3. Comparison of the Results and Discussion

In this paper, an analysis of different classifiers based on the different datasets shows the different characteristics of the different classifiers. The comparison of results can be found in Table 1.

| Methods  | Dataset | Accuracy | Recall | F Score |
|----------|---------|----------|--------|---------|
| GMM      | The emotion of Speech Data [30] | 81.25%   | NA     | NA      |
| SVM      | Amazon customers’          | 44.06%   | NA     | NA      |
Table 1 shows the Accuracy, Recall rate, and F score for different methods with different datasets. The first column presents the different methods to be compared; the second column gives the dataset that the method is tested on; the third column shows the accuracy of that method with a particular dataset; the Recall rate and F score are presented on the last two columns.

From Table 1, the results show that all of the classifiers perform well in the application of classification tasks. Especially, the Random Forest-based classifier has the highest accuracy in the classification of remote sensing. XGBoost, Naive Bayes, and GMM all have above 80% accuracy in their classification tasks. However, the SVM only has 44.06% accuracy during the text classification task. Obviously, multi-classification is an excellent challenge for machine learning classifiers. Especially, the more classes are, the more difficult the classification task would be. Therefore, the accuracy of the classifier is influenced heavily by the classification tasks.

4. Conclusion

In this paper, an analysis of classification performance with different methods has been given. In comparison, the differences among the performances of different classifiers are not significant. The performance, including accuracy, recall, and F score, is influenced by the classification tasks. If the classes in classification tasks are extensive, the accuracy would be very low. In this paper, the comparison results illustrated that the specific classification tasks are crucial for the performance of classifiers. The research of this paper can provide some suggestions for reference of the classification tasks.

References

[1] Kotsiantis S B, Zaharakis I, Pintelas P. Supervised machine learning: A review of classification techniques[J]. Emerging artificial intelligence applications in computer engineering, 2007, 160(1): 3-24.
[2] Murthy S K. Automatic construction of decision trees from data: A multi-disciplinary survey[J]. Data mining and knowledge discovery, 1998, 2(4): 345-389.
[3] Hunt E B, Marin J, Stone P J. Experiments in induction[M]. New York, Academic Press. 1966.
[4] Breiman L, Friedman J, Stone C J, et al. Classification and regression trees[M]. CRC press, 1984.
[5] Kononenko I. Estimating attributes: Analysis and extensions of RELIEF[C]//European conference on machine learning. Springer, Berlin, Heidelberg, 1994: 171-182.
[6] Breslow L A, Aha D W. Simplifying decision trees: A survey[J]. Knowledge engineering review, 1997, 12(1): 1-40.
[7] Littlestone N, Warmuth M K. The weighted majority algorithm[J]. Information and computation, 1994, 108(2): 212-261.
[8] Freund Y, Schapire R E. Large margin classification using the perceptron algorithm[J]. Machine learning, 1999, 37(3): 277-296.
[9] Good I J. Probability and the Weighing of Evidence[R]. London: C. Griffin, 1950.
[10] Nilsson, N.J. Learning machines. McGrawHill: New York. 1965.
[11] Torlay L, Perrone-Bertolotti M, Thomas E, et al. Machine learning–XGBoost analysis of language networks to classify patients with epilepsy[J]. Brain informatics, 2017, 4(3): 159-169.
[12] Huang Y, Englehart K B, Hudgins B, et al. A Gaussian mixture model based classification scheme for myoelectric control of powered upper limb prostheses[J]. IEEE Transactions on Biomedical Engineering, 2005, 52(11): 1801-1811.

[13] Witten I H, Frank E. Data mining: practical machine learning tools and techniques with Java implementations[J]. Acm Sigmod Record, 2002, 31(1): 76-77.

[14] Reynolds D A. Gaussian Mixture Models[J]. Encyclopedia of biometrics, 2009, 741: 659-663.

[15] Povey D, Ghoshal A, Boulianne G, et al. The Kaldi speech recognition toolkit[C]//IEEE 2011 workshop on automatic speech recognition and understanding. IEEE Signal Processing Society, 2011 (CONF).

[16] Narayanan A, Wang D L. Joint noise adaptive training for robust automatic speech recognition[C]//2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014: 2504-2508.

[17] Li J, Deng L, Gong Y, et al. An overview of noise-robust automatic speech recognition[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2014, 22(4): 745-777.

[18] Burges C J C. A tutorial on support vector machines for pattern recognition[J]. Data mining and knowledge discovery, 1998, 2(2): 121-167.

[19] Smola A J, Schölkopf B. A tutorial on support vector regression[J]. Statistics and computing, 2004, 14(3): 199-222.

[20] Herbrich R, Graepel T, Obermayer K. Large margin rank boundaries for ordinal regression[J]. Advances in large margin classifiers, 2000, 88(2): 115-132.

[21] Yu H. SVM selective sampling for ranking with application to data retrieval[C]//Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. 2005: 354-363.

[22] Hastie T, Tibshirani R. Classification by pairwise coupling[J]. Annals of statistics, 1998, 26(2): 451-471.

[23] Friedman J H. Another approach to polychotomous classification[J]. Technical Report, Statistics Department, Stanford University, 1996.

[24] Pal M. Random forest classifier for remote sensing classification[J]. International journal of remote sensing, 2005, 26(1): 217-222.

[25] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python[J]. the Journal of machine Learning research, 2011, 12: 2825-2830.

[26] Good I J. Probability and the Weighing of Evidence[R]. London: C. Griffin, 1950.

[27] Díaz-Uriarte R, De Andres S A. Gene selection and classification of microarray data using random forest[J]. BMC bioinformatics, 2006, 7(1): 1-13.

[28] McAuley J, Pandey R, Leskovec J. Inferring networks of substitutable and complementary products[C]//Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. 2015: 785-794.

[29] URL2, http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Coimbra#. Last Access (05.12.2018).

[30] Saritas M M, Yasar A. Performance analysis of ANN and Naive Bayes classification algorithm for data classification[J]. International Journal of Intelligent Systems and Applications in Engineering, 2019, 7(2): 88-91.