International Journal of Mathematics and
Computer Science, **17**(2022), no. 3, 995–1006

Hybrid Mathematical Model for Data Classification and Prediction: Case study COVID-19

Arwa Zabian[1], **Ahmed Zohair Ibrahim**[2]

[1]Department of Software Engineering
Faculty of Science and Information Technology
Jadara University
Irbid, Jordan

[2]Department of Computer Sciences
College of Computer and Information Sciences
Princess Nourah bint Abdulrahman University
Riyadh, KSA

email: arwa@jadara.edu.jo, azibrahim@pnu.edu.sa

(Received January 10, 2022, Accepted February 14, 2022)

Abstract

The increase of data availability has stimulated researchers to benefit from this data in predicting the hidden pattern for knowledge discovery. Data classification and machine learning algorithms are becoming important tools used in knowledge discovery. In this paper, we propose a hybrid classification model that combines some features and parameters from a probabilistic model and some other parameters from a divide and conquer model in a linear one. In our model, we generate a set of functions related to the number of attributes and the value of each attribute. Afterwards, these functions are reduced according to the number of classes needed. We test our model on collected data about symptoms in people infected with COVID-19 in England. Our simulation results show an accuracy rate in the range 50-80%. We expect to increase the accuracy rate if we increase the size of data used or we increase the number of attributes.

Key words and phrases: Data classification, unified mathematical model, probabilistic model, knowledge discovery, pattern prediction.
AMS (MOS) Subject Classifications: 68T07, 68T30.
ISSN 1814-0432, 2022, http://ijmcs.future-in-tech.net

1 Introduction

The availability and diversity of data produced every day have stimulated researchers to take advantage of that data to find the hidden patterns in it, to predict new behaviors, or to analyze natural phenomena. Pattern recognition is defined as the automatic discovery of regularities and irregularities in data for decision making, prediction, and classification. This kind of analysis needs tools and techniques to build a suitable model which represents the important aspects of the data.

Machine learning algorithms were used to provide a set of classification rules that made correct predictions in an independently chosen test set.

The goal of machine learning algorithms is to develop an efficient pattern recognition method that is able to scale with the size of the problem domain and data sets. Data analysis and machine learning algorithms can be used anywhere in any discipline like health, commerce, social sciences, image processing, optical character recognition (OCR), etc.

In health, machine learning algorithms are used to analyze and classify mammograms and X-ray images. In e-commerce, they are used to predict the best products, best customers or best prices. In social networking, machine learning algorithms are used to identify and analyze love and hate speeches and comments. Data analysis was used for many years to analyze the results obtained for a specific work, but what we try to do now is knowledge discovery, which means extracting useful information from huge, heterogeneous data. The performance of machine learning algorithms is considered based on classification accuracy that varies from a data set to another and from a field to another. The basic principle on which all machine learning algorithms work is transforming the input into a new form/space to facilitate the operation of classification or mapping with a low number of classes. In general, the size of the input is large and must be processed and mapped into 1-3 classes. The processing steps and input transformation vary from one algorithm to another. All machine learning algorithms are based on mathematical models. In this paper, we want to prove that a linear model can represent all the machine learning algorithms (those based on probabilities, linear, divide and conquer technique). For that, we have analyzed different algorithms (decision tree, Naïve Bayes) and we found that all have some things in common. We then used the concepts and parameters of a decision tree algorithm (that uses divide and conquer) in addition to those of Naïve Bayes algorithm (that is a probabilistic model) to construct a hybrid model that combines linearity and probability into a single model. Our model is a linear one that assigns to each attribute a weight, then generates a set of functions based on the number of attributes used. Afterwards, the number of functions is reduced according to the number of classes needed in a manner that each function represents one class only. As mentioned in [1], Naïve Bayes classifier, logistic regression, and decision tree are just alternative ways of representing a conditional probability distribution.

The rest of this paper is organized as follows: Section 2 represents an analysis of Naïve Bayes, decision tree and linear regression. In Section 3, we present some related work. In Section 4, we present our proposed model. Section 5 represents the analysis of the proposed model and a case study applied to COVID-19 data. Finally, there is a conclusion and a suggestion for future work.

2 Comparative analysis between decision tree algorithm and Naïve Bayes

A Decision tree algorithm is a simple classification technique that uses a divide and conquer method to construct a tree rooted at the attribute with the highest information gain where, in the leaves, we can find the desired classes. The best classification (path) is one with maximum information gain. The construction of a tree is based on impurity, meaning the degree of homogeneity in the data (single class or multiple classes), if all the data belongs to one class the Entropy will be equal to zero and no further classification can be done.
Entropy $= \sum_{j=1}^{n} -P(j) \log p(j)$,
where j is the number of classes and $p(j)$ is the probability of each class calculated as follows:
$P(j) = \frac{\text{the number of records for each class}}{\text{total number of records}}$.
Information gain $IG(A)$ for attribute A is a measure of the difference in entropy before deciding on the value of an attribute to the value after the decision is made [1].

$IG(S, A) = E(S) - \sum \frac{\text{\#of elements in the subset (t)}}{\text{\#of elements in all the dataset}} * E(t)$,
where $E(s)$ is the entropy of the dataset before a
split, $E(t)$ is the entropy of the attribute.

For example, consider a data set with three attributes A, B, C and two classes (+, -). Each attribute can have two values, 0 or 1. To construct the tree, we must follow the following steps:

1. Calculate the entropy for all the data.

2. Separate each value of each attribute and calculate the degree of impurity for each subset. Then calculate the entropy for each attribute.

3. Calculate the information gain for each attribute.

4. The attribute with maximum information gain will be the root (for example B).

5. Split the data set into two subsets based on the value of A and repeat the steps 1-4 to find the next level.

6. Repeat step 5 until arriving at a state where all the data belongs to one class, meaning the end of classification.

   The accuracy of prediction in decision tree algorithm varies in the range 82.73-98.11 given the comparison on forecast prediction presented in [2].

   A decision tree is a recursive algorithm that gives a good result in terms of predictions. The larger the size of the data set, the better the prediction accuracy. The main problem in the decision tree is its complexity in terms of calculations, where the number of splits increases with the number of attributes.

Naïve Bayes Algorithm is a machine learning algorithm based on Bayes theorem that is a statistical principle that combines prior knowledge of the class with new evidence gathered from data. A Bayesian classifier finds a probabilistic relationship between the attribute set and the class. Consider an attribute X and a class Y, P(Y) is the prior probability and P(Y | X) is the posterior probability of Y, which means given a known value of the attribute X we want to predict if X with that value belongs to the class Y [1]. During the training phase, we need to know posterior probability P(Y| X) for every combination of X and Y based on information gathered from the training data.

$P(Y|X) = \frac{P(X|Y)*P(Y)}{P(X)}$,

where $P(X|Y)$ is the conditional probability, meaning the probability that attribute X with a defined value belongs to the class Y.

The key objective in measuring the performance of this algorithm is the prediction accuracy.

Accuracy rate= $\frac{\text{number of correct predictions}}{\text{total number of predictions}}$

Accuracy rate $= \frac{f11+f00}{\text{total number of predictions}}$,
where $f11$ is the true predicted true, $f00$ is the false predicted false that means valid predictions.

Error rate $=\frac{\text{number of wrong prediction}}{\text{total number of predictions}}$.

The accuracy rate of Naïve Bayes varies in the range 70.32-95.63 as tested in [2].

From the comparison presented in [2], we conclude that the Naïve Bayes algorithm has better accuracy and is faster compared to the ID3 algorithm on the class studied.

Linear regression is the model used when the outcome or class is numerical and all the attributes are numeric. The idea is to express the class as a linear combination of the attributes with predetermined weights:

$X = w_0 + w_1a_1 + w_2a_2 + \ldots\ldots.w_ka_k$,
where X is the class, $a_1, a_2, \ldots$ are the attribute values and $w_0, w_1, \ldots$ are weights. A linear regression can be easily used for classification in domains with numeric attributes. It is an excellent and simple method for numeric prediction [1].

In our work, we will combine the calculations of the decision tree algorithm (Entropy IG) with the prior probability calculated in Naïve Bayes in linear functions to reduce the complexity of calculations in decision tree and benefits from the prior probability calculated in Naïve Bayes resulting in a hybrid model that combines the linear and probabilistic model in one function. The number of functions obtained depends on the number of attributes.

3 Related works

In [3], there is a proposed algorithm that combines Naïve bayes and Decision Tree for correct predictions of intrusion in a network. The problem in the actual intrusion detection systems is that the rate of false positives is high. The proposed algorithm uses Naïve Bayes to preprocess the data by calculating prior and conditional probabilities to classify and update training data based on it. Then, a decision tree algorithm is used to split the training data into sub-datasets based on the attribute with the highest information gain, recalculate prior and conditional probabilities for re-classifying each sub-data. This process continues until finding the correct classification. The proposed algorithm was applied and tested on KDD CUP1999 data set [4], for building a predictive model capable of distinguishing between intrusion and normal connections. The results show that, for detection rate normal access, the proposed algorithm performs slightly better (99.84 %) than Naïve Bayes (99.65 %) and ID3 (99.71 %) and performs equal to Naïve Bayes in terms of false-positives when tested using 19 attributes. The proposed algorithm outperforms decision tree and Naïve Bayes when the number of attributes is increased to 41 attributes. The paper concludes that the hybrid algorithm minimizes the rate of false positives.

The goal of the classification model is to obtain a classifier that minimizes the prediction error rate for unseen tested samples. In [5], a unified classification model is proposed, called Robust Classification Model (RCM) that handles optimization problems defined by uncertain inputs. RCM is a generalization of SVM (Support Vector Machine) [6], MPM (Minimax Probability Machine) [7], and FDA (Fisher Discriminant Analysis) [8]. The proposed model will help in clarifying the relationships among existing models and in finding a new classifier. In [9], an extended model of RCM that is based on APG (Accelerated Proximal Gradient) algorithm is proposed that can be applied to an arbitrary classification model without changing the algorithmic framework. In supervised learning, there is no single algorithm that can outperform all other algorithms in all cases, which means one algorithm can be best in one case and another algorithm is best in another case. The proposed algorithm FADC (Fast Accelerated Proximal Gradient) performs stably and can be applied not only to the unified classification model but for general convex composite optimization problems, the algorithm is proved to be stable and converges and is experimented using artificial data sets and benchmark data sets from LIBSUM data [10], and compared to LIBLINEAR algorithm [11], the numerical experiments show that FAPG is efficient for large data sets and runs faster than LIBLINEAR for large scale data set such that $n > 2000$.

In [12], three machine learning algorithms (Decision Tree-Naïve Bayes, K- Nearest Neighbors) are used to search for alternative architecture building designs in terms of energy performance. The three classifiers are used to classify 10 building designs. The results show that the decision tree is the fastest classifier in terms of classification time, Naïve Bayes outperforms Decision Tree and KNN algorithms in terms of prediction accuracy.

4 Proposed classifier

Data classification is the process of organizing data into categories/groups (similar or different). The classification process is divided into two phases: the first phase is the training phase where the classification model is built and the second phase is data classification based on the built model. In this paper, we are interested in proposing a hybrid classifier model that combines the probabilistic criteria used in Naïve Bayes algorithm with that used in the Decision Tree algorithm for linear functions. The number of functions obtained is related to the number of attributes used in the classification and the number of values for each attribute. Then, the number of functions is reduced to the number of classes needed. A Bayesian classifier assumes the independence among attributes, leading to simple and fast decision making. A Decision Tree classifier is a a classification method that uses a hierarchical model to classify the data where each accepted branch ends with a class and the internal nodes represent a test result for an attribute. The criteria used in Decision Tree algorithms for the construction of the tree is the information gain and entropy. In our model, to each attribute we assign a weight $W_{i,}$, where i is the number of the attribute.

$W_i = IG_i * P_i \qquad [1]$

IG_i is the information gain and P_i is the conditional probability. Our function takes the form:

$F_j = E(S) + \sum_{i=1}^{n} W_i, \quad (2)$

where $E(S)$ is the entropy of the total data, j is the number of functions obtained, and n is the number of attributes.

In our work, we assume that each attribute can take only two values (0, 1 / true, false / yes, no). For that if we use three attributes of data classification, we obtain 8 functions. Then, these 8 functions are reduced to 4 using a reduction method similar to Karnaugh map reduction used for simplification of logical expressions [13]. Afterwards, the 4 functions are reduced to the number of classes needed. If we want to classify the data into two classes, then we take the functions with maximum and minimum values. If we want to classify the data into 3 classes, then we take the logical or between the other remaining two functions.

4.1 Algorithm description

Before describing how our algorithm works and the resultant functions, we need to make some assumptions to use them in the calculation.

4.2 Assumptions

1. Each attribute X can have only two values, 0 or 1.

2. X_0 means the attribute takes the value 0, and X_1 means the attribute takes the value 1.

3. N is the number of classes. By assumptions, we classify the data into 2, 3 classes only.

4. $P(X_0|1)$ means the probability of an attribute to take value 0 and to belong to class 1.

   A similar definition is used for all the attribute values for all classes.

The classification method described in this paper passes in the following steps:

1. Dividing the data into data training and test data.

2. For data training, we calculate $E(S)$ as follows:
   $E(S) = -\sum_{i=1}^{n} P_i \log(P_i)$, where i is the number of classes and
   $P_i = \frac{\text{the probability of class}}{\text{total data size}}$.

3. For each attribute calculate the information gain as follows:

   $IG(X) = E(S) + \frac{p(X0)}{\text{total data size}} * E(X_0) + \frac{p(X1)}{\text{total data size}} * E(X_1)$,

   $E(X_0)$ is calculated as follows:

   $E(X_0) = \frac{p(X0|1)}{totalX0} * \log(p(X_0|1)) + \frac{p(X0|2)}{totalX0} * \log(p(X_0|2))$,

   $E(X_1)$ is calculated in the same manner as $E(X_0)$.

4. To generate our functions, consider three attributes A, B, C with each attribute having two values. That generate 8 possible functions given the following map:

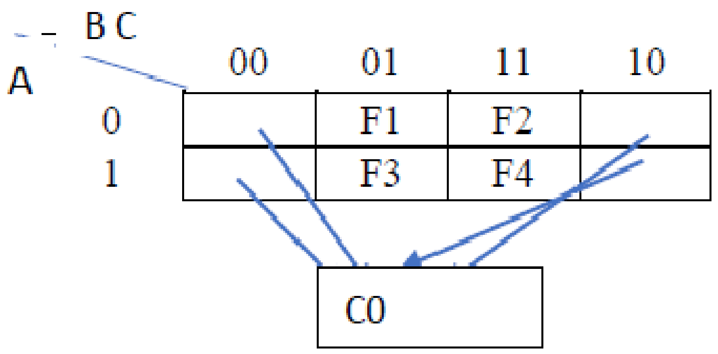

Figure.1: functions map

The 8 functions are reduced to the 4 functions $F1, F2, F3$, and $F4$ as in Figure 1, as follows:

$F1 = P(C_0) + IG(A_0) * P(A_0) + IG(B_0)P(B_0) + IG(C_1) * P(C_1)$

$F2 = P(C_0) + IG(A_0) * P(A_0) + IG(B_1)P(B_1) + IG(C_1) * P(C_1)$

$F3 = P(C_0) + IG(A_1) * P(A_1) + IG(B_0)P(B_0) + IG(C_1) * P(C_1)$

$F4 = P(C_0) + IG(A_1) * P(A_1) + IG(B_1)P(B_1) + IG(C_1) * P(C_1)$

If we have two classes, then consider the maximum and minimum functions. If we have three classes, then we add the OR of the two other functions.

5. Apply the maximum and minimum function of the test data to find the classification results.

5 Experimental results

We have tested our model on a real dataset from a COVID-19 infection survey conducted by the Office for National Statistics in England. The data was published on February 9, 2021.
Data is available via:
infection.survey.analysis@ons.gov.ukinfection.survey.analysis@ons.gov.uk

Data is collected as a sample of 246 people in total, tested positive with cough, fever, and shortness of breath. The statistics refer to infections reported in the community by which we mean private households, excluding infections reported in hospitals, care homes, or other institutional settings. The infection ratio is 38.8% which means 95 cases from our data were infected.

Data is converted to a new form to be presented in our system as follows: three attributes are used to describe the patient status: fever (represented as variable A), shortness of breath (represented as variable B), and cough (represented as variable C). The symptom is presented when the variable takes the value 1; otherwise, the variable takes value 0. The state of the patient is classified as infected (1) or not infected (0). The infection ratio is considered as 39%, the percentage of positive results without fever is 80%, and without shortness of breath is 80% and without cough is 63%.

60% of data is used for training (total 147) and 40% of data is used for test data (total 99%)

Our results show that true positive means infected with fever, cough, and shortness in breath will correspond to the maximum function that is F4. However, false-negative corresponds to not infect that represented by the minimum function; that is, F1. The accuracy rate on this data is variable, it can be from 50% in most cases and can reach up to 80%. We have changed the attributes taken from the same dataset like fever, fatigue, and headache. We have obtained similar results that indicate F4 is infected and F1 is not infected with the same accuracy rate. We expect that the accuracy rate of our classifier will increase when the size of data is increased and the number of attributes is increased.

6 Conclusion and future work

In this paper, we have proposed a mathematical model that combines properties from different machine learning algorithm types to obtain a hybrid linear model, which is easy to understand and apply. We have applied our work on real datasets collected for COVID19 symptoms and we have obtained results as expected, that the maximum function will indicate one class and the minimum function will indicate the other class. The accuracy rate on our data test is about 50%. We expect to increase the accuracy rate when the size of data is increased or when the type of data collected is changed. In our future work, we intend to collect datasets from different disciplines like health or social network to find the behavior of our model on different types of datasets and to increase the accuracy rate.

References

[1] Ian H. Witten, Eibe Frank, Mark A. Hall, Christopher Pal, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2017.

[2] I. Sarwani, Comparative Analysis of ID3 and Nave Bayes Algorithm on Stock Market Prediction, International Journal of Computer Applications, **143**, no. 2, (2016).

[3] Dewan Md. Farid , Mohammad Zahidur Rahman, Harbi Nouria, Combining Naïve Bayes and Decision Tree for Adaptive Intrusion Detection, International Journal of Network Security & its Application, **2**, (2010).

[4] The KDD Archive KDD99 CUP data set, (1999).

[5] Akiko Takeda, Hiroyuki Mitsugi, Takafumi Kanamori, A unified Classification Model Based on Robust Optimization, Neural computation, **25**, no. 3, (2013), 759–804.

[6] B. Schölkopf, Alex Smola, R. C. Wiliamson, P. Bartlett, New support Vector Algorithm, Neural Computation, **12**, no. 5, (2000), 1207–1245.

[7] Gert R. G. Lanckriet, Laurent El Ghaoui, Chiranjib Bhattacharyya, Michael I. Jordan, A Robust Minimax Approach to classification, Journal of Machine Learning Research, **3**, (2002), 555–582.

[8] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1990.

[9] Naoki Ito, Akiko Takeda, Kim-Chuan Toh, A Unified Formulation and Fast Accelerated Proximal Gradient Method for classification, Journal of Machine Learning Research, **18**, (2017), 1–49.

[10] Chih-Chung Chang, Chih-Jen Lin, LIBSVM: A Library For Support Vector Machines, ACM Transactions on Intelligent Systems and Technology, **2**, no. 3, (2011), 1–27.

[11] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hseih,Xiang-Rui Wang, Chih-Jen Lin, Liblinear: A Library For Large Linear classification, The Journal of Machine Learning Research, **9**, (2008), 1871–1874.

[12] Ahmad Ashari, Iman Paryudi, A. Min Tjoa, Performance Comparison Between Naïve Bayes, Decision Tree, and K-Nearest Neighbor in Searching Alternative Design in An Energy simulation, International Journal of Advanced Computer and Applications, **4**, no. 11, (2013), 33–39.

[13] Tony Kuphaldt, Introduction to Karnaugh Mapping, Lessons in Electric Circuits, 2007, Available at:
https://www.allaboutcircuits.com/textbook/digital/chpt-8/introduction-to-karnaugh-mapping/