Journal of Engineering and Applied Sciences

MEDWELL PUBLICATIONS
SINCE 2000

A New Approach for Classification of Network Data using One Class SVM

[1]N. Raghavendra Sai and [2]K. Satya Rajes
[1]*Department of Computer Science, Bharathiar University, Tamil Nadu, India*
[2]*Department of Computer Science and Engineering, SRR and CVR Degree College, Vijayawada, AP, India*

Key words: SVM, recognizes, algorithm, pattern, high-dimensional

Corresponding Author:
N. Raghavendra Sai
Department of Computer Science, Bharathiar University, Tamil Nadu, India

Page No.: 3146-3151
Volume: 15, Issue 17, 2020
ISSN: 1816-949x
Journal of Engineering and Applied Sciences
Copy Right: Medwell Publications

Abstract: One class classification recognizes the target class from all other classes using only preparing data from the target class. One class order is suitable for those situations where exceptions are not spoken to well in the training set. One-class learning or unsupervised SVM, aims at isolating data from the origin in the high-dimensional, indicator space (not the original predictor space) and is an algorithm used for outlier detection. Support vector machine is a machine learning method that is widely used for data examining and pattern recognizing. Support Vector Machines (SVMs, also Support Vector Networks) are supervised learning models with related learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. In this study, we will review the difference between both these classes.

INTRODUCTION

Classification and relapse are the two most normal types of models fitted with administered preparing. At the point, when the model must pick which of a few discrete classes the example information have a place with grouping is utilized. Likewise, when the model must compute a numeric output from the input data, regression is used. Classification is used when the output is discrete or categorical and regression is used when the output is continuous or numeric. It quickly becomes more complex than this simple case. Many models such as support vector machines or Generalized Linear Models (GLMs), only support binary classification. They can only directly classify between two discrete classes. Yet, these models are often used for many more than two classes. Similarly, neural networks and linear regression only directly support regression. In this study will look at three distinct applications of supervised learning:

- Binary classification
- Multi classification
- Regression

The exact means by which several models support these three will be discussed. This study will specifically examine the following models:

- Generalized linear regression (GLM)
- Linear regression
- Neural networks
- Support vector machines
- Tree-based models

SIMPLE REGRESSION

Linear regression is one of the most basic, yet still useful, types of model. One representation of the linear regression formula is given by Eq. 1.

Equation 1: Linear regression:

$$\hat{y} = \beta_1 x_1 +, ..., +\beta_n x_n \qquad (1)$$

The output prediction ŷ(y-hat) is calculated by the summation of multiplying each input element (x) by a corresponding coefficient/weight (β). Training the linear regression model is simply a matter of finding the coefficient values that minimize the difference between y and the actual y. It is very common to append a constant value, typically 1, to the input vector (x). This constant allows one of the coefficient (β) values to serve the y-intercept. The returned value is numeric a regression was performed. Common examples of linear regressions derive coefficients to determine shoe size, based on height, or a person's income based on several other numeric observations. Linear regression is best at modelling linear relationships. For nonlinear relationships, a neural network or Generalized Linear Model (GLM) might be used. A single neuron in a neural network is calculated by Eq. 2.

Equation 2: GLM or single neuron calculation:

$$\hat{y} = f(x, w) = \Phi(x_1 w_1 +, ..., +x_n w_n) \qquad (2)$$

The output from a single neuron is very similar to the linear regression. An input/feature vector (x) is still the primary in-put. However, neural network terminology usually refers to the coefficients (β) as weights (w). Usually a constant input term is appended, just like linear regression. However, neural networks terminology refers to this weight as a bias or threshold, rather than the y-intercept. The entire summation is passed to a trans-fer, or activation function, denoted by Φ. The transfer function is typically sigmoidal, either logistic or the hyperbolic tangent. Newer neural networks, particularly deep neural networks will often use a Rectifier Linear Unit (ReLU) as the transfer function. Neural networks are typically made of layers of neurons such as Fig. 1.

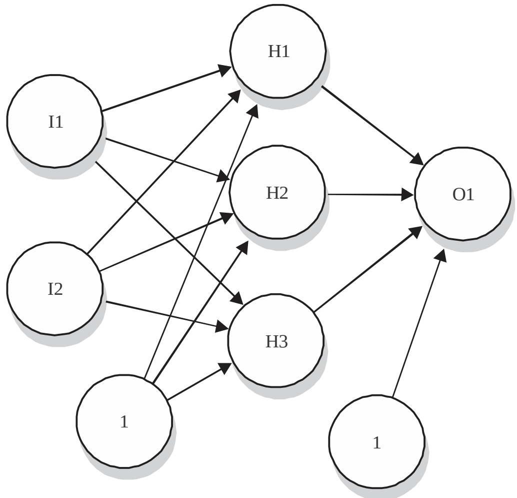

Fig. 1: Neural network

The output from a neural network is calculated by first applying the input vector (x) to the input neurons in this case I1 and I2. A neural network must always have the same number of input neurons as the vector size of its training data (x). Next, calculate the values of each hidden neuron H1, H2, etc., working forward until the output neuron(s) are calculated. The output for a GLM is calculated exactly the same as a single neuron for a neural network. However, the transfer/activation function is referred to as a link function. Because of this, a neural network can be thought of as layers of many GLMs. The error for a neural network or GLM can be thought of as the difference between the predicted output (ŷ) and the expected output (y). A common measure of the error of neural networks and sometimes GLMs is Root Mean Square Error (RMSE), the calculation for which is shown by Eq. 3.

Equation 3: RMSE:

$$E = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_t - y_t)^2}{N}} \qquad (3)$$

The constant N represents the number of items in the training set. RMSE is very similar to the standard deviation calculation used in statistics. RMSE measures the standard deviation from the expected values.

BINARY CLASSIFICATION

Binary classification is when a model must classify the input into one of two classes. The distinction between regression and bi-nary classification can be fuzzy. When a model must perform a binary classification, the model output is a number that indicates the probability of one class over the other. This classification is essentially a regression on the probability of one class vs. the other being the correct outcome! For many models, binary classification is simply a special case of regression.

A popular form of binary classification for the GLM model is logistic regression where the link function is the logistic function. If the GLM, using logistic regression, is to classify more than two categories, a special voting arrangement must be used. This is discussed later in this article. Binary classification provides a number that states the probability of an item being a member of a category. However, this brings up the question of what a sufficient probability is for classification. Is a 90% probability enough? Perhaps a 75% probability will do. This membership threshold must be set with regard to the willingness to accept false positives and false negatives. A higher threshold decreases the likelihood of a false positive, at the expense of more false negatives. A Receiver Operating Characteristic (ROC) curve is often used, for binary classification, to visualize the effects of setting this threshold.

As the threshold is set more or less restrictive, the true positive rate and false positive rates change. A threshold is represented as one point on the curved line. If the true positive rate is very high, so will be the false positive rate. The reverse also holds true. Sometimes it is valuable to measure the effectiveness of the model, independent of the choice of threshold. For such cases, the Area Under the Curve (AUC) is often used. The larger the area below the curve, the better the model. An AUC of 1.0 is a perfect but highly suspicious, model. Nearly "perfect" models are rare and usually indicate over fitting. Calculating the AUC can be complex and often employs similar techniques to integral estimation. It is rare that AUC calculation can be performed using definite symbolic integration.

MULTIPLE CLASSIFICATION

AUC curves are only used for binary classification. If there are more than two categories, a confusion matrix might be used. The confusion matrix allows the analyst to quickly see which categories are often mistaken for each other. A confusion matrix for the classic iris dataset is shown by Table 1.

The iris dataset is a collection of 150 iris flowers, with four measurements from each. Additionally, each iris is classified as one of three species. This dataset is often used for example classification problems. The confusion matrix shows that the model in question predicted Setosa correctly 46 times but misclassified a Setosa as Versicolor once and Virginica three times. A strong model will have its highest values down the northwest diagonal of a confusion matrix. It is important to understand how a model reports the prediction for a multiclassification. A model will report a vector for the input data. The model might report 90% Setosa, 7% Versicolor and 3% Virginica for a set of flower measurements that the model felt was likely Setosa. In this case, the model would return the following vector: (0.9, 0.07, 0.03). This is very different from the typical multiple choice question format that might be seen on an actuarial exam. For such an exam, the answer must be chosen as either A, B, C or D. The model has the advantage of being able to choose its confidence in each of the possible choices. Classification problems are often numerically evaluated using the multiple log loss as shown by Eq. 4.

Table 1: Multiple-classification

	Predicted		
Actual	Setosa	Versicolor	Virginica
Setosa	46	1	3
Versicolor	2	46	1
Virginica	1	1	48

Equation 4: Multi-log loss:

$$E = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\log\left(\hat{y}_{i,j}\right) \tag{4}$$

The constant N represents the number of training set items and M represents the number of classes. Like previous equations in this study, y represents the model prediction and y represents the expected outcome. The lower the log loss, the better. To explain how Eq. 4 works, think of the multiple choice exam previously mentioned. If the correct answer for a particular question was A and the model had given a 0.97 probability to A, then -log(0.97) points would be added to the average score. Log loss can be harsh. Predicting 1.0 correctly will add zero log-points to the error but predicting 1.0 incorrectly will give an infinitely bad score. Because of this, most models will never predict 1.0.

MULTIPLE REGRESSION

Just like multiple classification, multiple regression also exists. Neural networks with multiple outputs are multiple regression models. Usually, a neural network with multiple outputs is used to model multiple classification. This is how neural networks which are inherently regressive are made to support classification. A binary classification neural network simply uses a single output neuron to indicate the probability of the input being classified into one of the two target categories. For three or more categories, the output neurons simply indicate the class that has the greatest probability. Figure 2 shows a multiple out-put neural network.

Because a binary classification neural network contains a single output neuron and a three or more classification network would contain a count equal to the number of classes, a two-output neuron neural network is rarely used. While a neural network could be trained to perform multiple regressions simultaneously, this practice is not recommended. To regress multiple values, simply fit multiple models.

SOFTMAX CLASSIFICATION

For classification models, it is desired that the probabilities of each class sum to 1.0. Neural networks have no concept of prob-ability. The output neuron, with the highest value is the predicted class. It is useful to

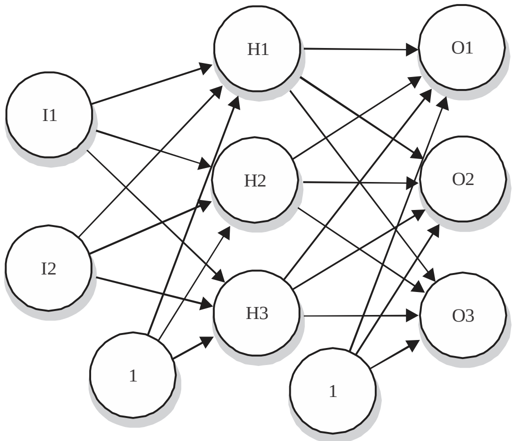

Fig. 2: Multi-output neural network

balance the output neurons of neural networks and some other models, to mirror probability. This is accomplished with the softmax as shown by Eq. 5.

Equation 5: Softmax:

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}, \; j = 1, ..., K \quad (5)$$

Softmax can be used to transform any multi-output regression model to a classification model. Most neural network-based classifications make use of the softmax function. It is based on the logistic function and provides the same sort of squashing effect at the extremities. The softmax function is very similar to simply summing the output neurons and balancing each neuron to be-come the proportion of this summation. This approach is often called normalization or simply hardmax. The softmax softens this approach and usually makes the probabilities more realistic.

Voting: Many models can only function as binary classifiers. Two such examples are GLMs and support vector machines (SVM). Not all models have this limitation, any tree-based model can easily classify beyond two classes. For a tree, the leaf-node specifies the class. It is very easy to convert any binary classifier into a three or more classifier. Figure 3 shows how multiple binary classifiers could be adapted to the iris dataset for classification.

Essentially, a classifier is trained for each of the output categories. For the iris dataset, three additional datasets are created, each as a binary classifier dataset. The first dataset would predict between Setosa and all other classes. Each class would have a dataset and model that predicts the binary classes of that category and all others. When using such a model, the input data would be presented to each of the three models and the data would be classified as belonging to the class that predicted the highest probability. Like a multi-output neural network, it would be helpful if the probabilities of each class summed to 1.0. This can be accomplished with the softmax function. By using multiple binary classifiers and a softmax, any binary classifier can be expanded beyond two classifications.

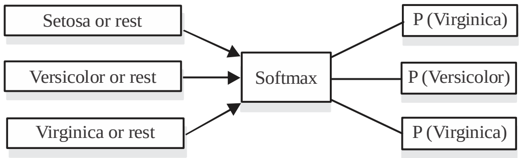

Fig. 3: Model voting

One class classification: Overview One-class classification has also been known as novelty or outlier detection[1]. Different from normal classification, it samples data from only one class, called the target class, are well characterized while there are no or few samples from the other class (also called the outlier class). The one-class characterization issue contrasts in a single basic angle from the regular arrangement issue. In one-class order it is expected that exclusive data of one of the classes, the objective class is accessible. This implies just illustration objects of the objective class can be utilized and that no data about alternate class of exception objects is available. The limit between the two classes must be evaluated from information of just the ordinary, certifiable class. The errand is to characterize a limit around the objective class with the end goal that it acknowledges however much of the objective protests as could be expected while it limits the possibility of tolerating exception objects. A classifier, i.e., a capacity which yields a class name for each info question, can’t be built from known guidelines. In this way, in design acknowledgment or machine learning, one tries to construe a classifier from a (constrained) arrangement of preparing cases. The utilization of cases in this manner hoists the need to expressly express the tenets for the order by the client. The objective is to get models and taking in guidelines to gain from the illustrations and foresee the names of future articles. The objective of the one-class classification is to recognize an arrangement of target items and all other conceivable articles. It is for the most part used to identify new protests that take after a known arrangement of items. At the point when another question does not look like the information, it is probably going to be an anomaly or a curiosity[1]. When it is acknowledged by the information portrayal, it can be

Table 2: Difference between one class and SVM classification

One class classification	SVM classification
One class contains data from only one class, target class	SVM contains data of two or more classes
Goal is to create a description of one class of objects and distinguish from outliers	Goal is to create hyperplane with maximum margin between two classes
Decision boundary is estimated in all directions in the feature space around the target classes	Hyperplane is created in between datasets to indicate which class it belongs to
Software finds an appropriate bias from such that outlier fraction of the observations in the training set	Software attempts to remove 100* outlier fraction% of observations when the optimizations algorithm converges
One class classifier is traditional classifier	SVM classifier is linear classifier
Used for outlier detection novelty detection	Used for classification regression
Less parameters, less training data	More parameters, more training data
Term coined by Moya and Hush	Term coined by Vladimir Vapnik

utilized with higher condense in a resulting grouping Different techniques have been created to make an information depiction. As a rule the likelihood thickness substantial number of tests to beat the scourge of dimensional. Different procedures than assessing a likelihood thickness appraise exists. It is conceivable to utilize the separation to model or just to gauge the limit around the class without assessing a likelihood thickness.

One class classification methods: There are four simple models[2], the support vector data description, k-means clustering, k-center method and an auto-encoder neural network. Here, a descriptive model is fitted to the data and the resemblance to this model is used. In the SVDD, a hyper sphere is put around the data. By applying the kernel trick the model becomes more flexible to follow the characteristics in the data. Instead of the target density the distance to the center of the hyper sphere is used. In the k-means and k-center method the data is clustered and the distance to the nearest prototype is used. Finally, in the auto-encoder network the network is trained to represent the input pattern at the output layer. The network contains one bottleneck layer to force it to learn a (nonlinear) subspace through the data. The reconstruction error of the object in the output layer is used as distance to the model. The utilization of an information area depiction technique is propelled by the help vector machine by called the Support Vector Domain Description (SVDD)[3]. This strategy can be utilized for curiosity or exception location. A roundly formed choice limit around an arrangement of items is developed by an arrangement of help vectors portraying the circle limit. It has the likelihood of changing the information to new component spaces without much additional computational cost. By utilizing the changed information, this SVDD can get more adaptable and more exact information portrayals. The blunder of the main kind, the part of the preparation objects which will be rejected can be evaluated promptly from the depiction without the utilization of an autonomous test set which makes this technique information productive. The support vector domain description is contrasted and other exception identification strategies on genuine information.

SVM classification: A help vector machine[4, 5] develops a hyperplane or set of hyperplanes in a high-or limitless dimensional space which can be utilized for characterization, relapse or different undertakings. Given an arrangement of preparing cases, each set apart as having a place with one of two classifications, a SVM preparing calculation constructs a model that doles out new cases into one classification or the other, making it a non-probabilistic twofold straight classifier. A SVM demonstrate is a portrayal of the cases as focuses in space, mapped with the goal that the cases of the different classes are isolated by an unmistakable hole that is as wide as could be allowed. New cases are then mapped into that same space and anticipated to have a place with a class in view of which side of the hole they fall on. Naturally, a great partition is accomplished by the hyper plane that has the biggest separation to the closest preparing information purpose of any class (so called useful edge), since by and large the bigger the margin the lower the generalization error of the classifier (Table 2).

CONCLUSION

Classification and regression are the two most common formats for supervised learning. As this article demonstrated, models have entirely different approaches to implementing classification and regression. Often the software package will take care of these differences. However, understanding the underpinnings of the models can be useful. For example, if a model were trained to recognize a large number of classes, then a GLM or SVM might not be a good choice.

If there were 10,000 possible outcome classes, a binary-only classifier would need to create a voting structure of 10,000 models to vote upon each classification. A large tree/forest or neural network might be able to more effectively handle such a problem. I conclude that this paper shows the difference between one class SVM and SVM classification and results show that SVM classification is more efficient than one class classification. Results shows as the number of parameters increases in One Class SVM, the performance decreases of one class classifiers. SVM classifiers, on the other hand, display stable performance and hence SVM classifiers are more appealing.

REFERENCES

01. Tax, D.M.J., R.P.W. Duin, N. Cristianini, J. Shawe-Taylor and B. Williamson, 2001. Uniform object generation for optimizing one-class classifiers. Mach. Learn. Res., 2: 155-173.
02. Wang, Q., L.S. Lopes and D.M. Tax, 2004. Visual object recognition through one-class learning. Proceedings of the International Conference on Image Analysis and Recognition, September 29-October 1, 2004, Springer, Porto, Portugal, pp: 463-470.
03. Scholkopf, B., 1997. Support vector learning. M.Sc. Thesis, R. Oldenbourg Verlag, Munich, Germany.
04. Cortes, C. and V. Vapnik, 1995. Support-vector networks. Mach. Learn., 20: 273-297.
05. Tax, D.M.J., 2002. One-class classification: Concept learning in the absence of counter-examples. Ph.D. Thesis, Delft University of Technology, Delft, Netherlands.