Applying Genetic Algorithm and Bimodal distribution removal to improve classification problem

Xiaosong Li
[1] Research School of Computer Science
Australian National University
U6108106@anu.edu.au

Abstract. This article mainly introduces two ways to improve the accuracy of classification. Based on Breast Cancer Data Set, this classifier could predict whether breast mass is malignant or not according to diagnosis situation. Not only applying effective neural network, but this article also demonstrates outlier removal algorithm to solve real data, which sometimes contain noise in training set. Additionally, Genetic Algorithm is applied to training process to do feature selection. Thus, outlier removal could remove some redundant patterns, and EA could select the subset of features. From results, we can see that above optimization could improve training process. In this article, I will discuss ways of optimizing training algorithm and improvement of these operations. Finally, I compared result with baseline according to accuracy in test set.

Keywords: neural network, classification, outliers, bimodal distribution, variance, loss, feature selection, genetic algorithm.

1 Introduction

Classification problem is common in academic fields and our life. Accurate classification model and efficient training process can be applied to solve many problems. On the one hand, in many situations, real world dataset used to train classification model always contains noise data, called the outlier. My first topic is how to reduce negative effect of the outlier and to improve accuracy in classification problem when meeting noisy data. On the other hand, in some dataset, redundant features and the complex neural network will cost too long computation times. To reduce its complexity, the model should apply Genetic Algorithm to select necessary features. My second part is feature selection. Above two topics all focus on improving the model by optimizing input. In this section, I will introduce the background of these topics and main contribution of my algorithm in the classification problem.

1.1 Background

1.1.1 Outliers detection

Some researchers explained that the limitation of non-parametric learning is dilemma between bias and variance [4]. Namely, preprocessing data before estimator in model might lead convergence to an incorrect solution, which is known as bias. In other word, how to examine outlier data points is a challenging problem.
Researchers mainly introduced a method, called Bimodal Distribution Removal (BDR), to solve variance influence from and to avoid convergence to incorrect solution [4]. They observed the change of frequency distribution for every 50 epochs in training set, to investigate the behavior of outliers in the training set during training process. From frequency distribution after 200 epochs, they found that some points still held high error. To reduce time complexity, they held similar measures to observe outliers, called variances. When variances drop to below 0.1, removal of outliers can begin. Additionally, they use two steps to determine which patterns are outliers in current epochs. Firstly, mean of errors for all patterns in training set is calculated to divide training set into two parts. In one part, all patterns errors are larger than mean value. Secondly, in this part, errors of some patterns are larger than threshold, which is related to mean and standard deviation of this part, will be considered as outlier points. Removal would be repeated every 50 epochs until variances of all training set are less than 0.01.
As mentioned in above part, this article has some limitations: Firstly, as for complex training set, the removal beginning time introduced in this article might be not reasonable. For example, 200 – 500 epochs are not reasonable to some special datasets, which contains too many noise patterns. Secondly, as for how to measure errors for each pattern, method in this article could not be applied to many situations, such as binary classifications. Thirdly, its evaluation is lack of explanations of accuracy in test dataset.

1.1.2 Genetic Algorithm

Features selection skills are used for following reasons. Firstly, simplification of model could reduce model's training time. Secondly, not too high dimensions enhance model's generalization by reducing overfitting.
As for this field, genetic algorithm belongs to branch of evolutionary algorithm, to select the more relevant features and reduce the use of redundant features for modelling [5]. Genetic algorithm could be applied to do feature selection commonly relying on its bio-inspired operations, called mutation, crossover, and selection. In [8], author combined accuracy with cost, as fitness function, and explained that multiple-criteria optimization problem could apply genetic algorithm to explore subset solution.

1.2 motivation and main contribution

To run my neural network model, I selected the dataset *Wisconsin Diagnostic Breast Cancer (WDBC)*, in this dataset, features are computed from a digitized image of breast mass. 30 numerical features to describe this breast mass, and one feature represent ID [1]. Label of this dataset represent two classes, that mass is malignant, or this mass is benign. This dataset is selected due to following reasons. Firstly, this dataset has 569 instances. As for binary classification problem, number of instances and numbers of features are large enough. Secondly, its numerical features do not need special encoding method. Thirdly, this dataset might have redundant features and some noise patterns. As for features, some information recorded as feature are not necessary for predict in medical dataset. As a cancer, some noise data might be stored in dataset due to some sudden situation. Thus, this dataset is suitable for BDR task and feature selection task. Additionally, investigation about which features are key information to predict whether mass is malignant, is necessary in this field.
My own contribution in this article mainly has four parts: (1) Use neutral network to do accurate classification task. (2) According to output performance to eliminate noise in input level and define an error measure for each pattern in output level. (3) Applying Genetic Algorithm to do feature selection. (4) Discussion result and analysis about feature selection.

2 Method and Algorithm

In this section, I will introduce my own methods to improve the performance of classification model. My introduction mainly focuses on my special design in model and algorithm, including three parts: (1) architecture of my neural network, including number of hidden layer and neurons, activation function, loss function, optimizer. (2) introduction about Bimodal distribution removal, including when model begins to remove outliers, and how to do removal. (3) introduction about feature selection, including Genetic Algorithm process. Additionally, in the last part of this section, I will introduce evaluation method for my model.

2.1 Architecture of neural network

As for structure of neural network, because this classification problem is not too complex, the number of hidden layer should be defined one or two layers. Number of inputs is equal to the number of features. Two output neurons return two dimensions vector, then using max function to classify each pattern to one class. In hidden layer, I applied ReLu activation function. In last layer, I applied *CrossEntropyLoss* loss function, and SGD optimizer.

2.1.1 Rectified Linear Units

In my neural network, ReLu is applied to be activation function. Rectified Linear Units function is defined as $\boldsymbol{h=max(0,a)}$ where $\boldsymbol{a=Wx+b}$. it means that if input less than 0, output is 0, and its gradient is a constant.
Compared to sigmoid function and tanh function, ReLu has following four advantages:

(1) it is more similar as biological thinking.
(2) In hidden layer, only about half of neurons are under activation status, others' outputs are equal 0.
(3) This activation has efficient gradient propagation, namely, its gradient is easily to get. ReLu avoid gradient vanishing problem during.
(4) Unlike sigmoid function, its output has sparsity, if output is less than 0, output will be equalized to 0. Compared to sigmoid function, it makes training process more quick convergence and efficient.

Based on above advantages, I selected ReLu activation in hidden layer. Because its special sparsity construction, its learning rate might be small after a certain number of epochs, to avoid too large average magnitude of updates from SGD optimizer step. It might push entire distribution of inputs of activation function into negative region.

2.1.2 SGD optimizer

Stochastic gradient descent is a method which update parameter based on one batch data, rather than all dataset. It reduces the time complexity and accelerate convergence of model. Dataset in this article is to predict breast cancer, based on diagnostic data set. Thus, I have some redundant data. As a result, if considering batch as basic unit to traverse all dataset, its update will be more efficient than using dataset as a whole input. My dataset might contain similar data, so SGD will be an efficient optimizer in my neural network model.

2.2 Introduction about Bimodal Distribution Removal

2.2.1 Error measure

In this section, I will introduce how to get error of each pattern. In classification, how to define error for each pattern is a problem. In [4], explanation about error of each pattern is not specific. As they explained, they random create 50 sets of 70 patterns and calculate subset's error to remove pattern. As for big dataset, this method is not accurate to select outliers. Thus, I use my own method to get error for each pattern. After a certain number of epochs, the *torch.nn.CrossEntropyLoss()* loss function could be applied to get error for all patterns . This loss function is useful when training a classification problem. The loss could be described as formula (1):

$$loss(\mathrm{x}, \mathrm{class}) = -log\left(\frac{exp(x[class])}{\sum_j exp(x[j])}\right) = -x[class] + log\left(\sum_j exp(x[j])\right) \quad (1)$$

class =0,1; x is a two dimensions vector

The output of *CrossEntropyLoss()* could represent average of difference between target label and predicted label among all patterns. If changing its parameter *reduce=False*, it returns a loss of per batch. This loss could be defined as error of each pattern, because if one pattern is assigned to wrong class, its loss value will larger than right classification situation.
Additionally, another loss function could be applied to measure error of each pattern, *torch.nn.BCELoss()* . This loss function is similar as above loss function, if the parameter *reduce* is changed to *False*, it also returns loss of each pattern. Its loss could be described as:

$$l_n = -w_n[y_n.logx_n + (1 - y_n).log(i - x_n)] \quad (2)$$

w = weights of loss of each pattern, x and y has same dimension.

From this formula (2), we can see that if x (predicted label) is approximate to y (actual label), its output is close to 0. If this pattern is aligned to wrong class, its output will close to 1. Thus, its output could represent its loss.

2.2.3 Pattern Removal

In above section, I have introduced how to decide the beginning time and how to measure error for each pattern. This section mainly explains how to define outliers and remove them. During removal process, patterns should be removed slowly, because errors of some patterns would be not extremely large or extremely small. These midrange errors could be learnt by network eventually. Thus, midrange errors should be remained in training set [4]. They explained two steps need to be implemented to remove actual outliers. Firstly, calculate the mean of errors of all the patterns in training set $\bar{x}$. According to requirement of pattern removal, this $\bar{x}$ is very small because dataset has been learnt by model more than 200 epochs (explained in above section). The whole dataset will be divided by two subsets. All patterns have larger errors than $\bar{x}$ will be aligned in subset x_1. Secondly, in subset x_1, means $\bar{x}_1$ and standard deviation σ_1 can be calculated. Then we defined outlier's error:

$$\text{error} \geq \bar{x}_1 + \boldsymbol{\alpha} * \sigma_1 \qquad 0 \leq \boldsymbol{\alpha} \leq 1 \quad (3)$$

According to formula (3), if $\boldsymbol{\alpha}$ is too small, pattern removal will be too quick. As a result, removing too much data, which including some normal data, will reduce model's generalization. If $\boldsymbol{\alpha}$ is too large, removal process cannot remove noise, because its threshold is too high.
The ending time of removal is that the variance of all errors v_{ts} less than a constant (normally 0.01). Satisfying ending requirement means that most of outliers have been removed and variance of errors is small enough.

2.3 Introduction about Genetic Algorithm

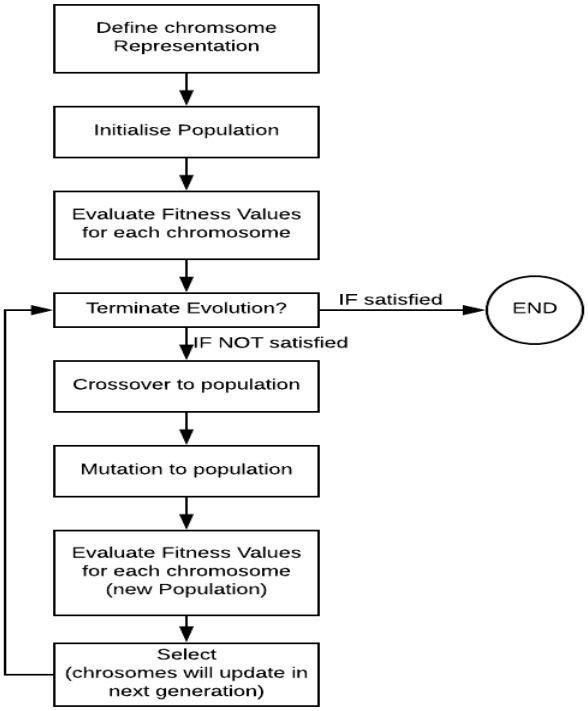

Figure 1: process of Genetic Algorithm [6]

Figure 1 shows the whole process of Genetic Algorithm. In the first step, chromosome is defined as an individual, which contains gene set. Size of chromosome is equal to numbers of feature in dataset. In my dataset *Wisconsin Diagnostic Breast Cancer (WDBC)*, it has 32 features to predict label. Thus, length of chromosome is 32. Values of genes are 0 or 1. The genes' value represent which features we need to select. Each gene represents one features. If gene's value is equal 1, its related features will be selected as input. If gene is equal to 0, its related feature will be removed. Thus, according to one chromosome, subset of features could be selected to input neural network. In the second step, population which contains 100 chromosomes, are generated randomly. The third step is getting chromosome's fitness in population. This part is important to link Genetic Algorithm and neural network. After finishing training process and test process relying on current subset of features, model will get accuracy of model in test set. This accuracy can be defined as fitness of this chromosome, which determines current subset of features. High value of fitness represent that this subset of features is more efficient than others subset. In step 4, definition of termination requirement determines whether model need next generation or not. In step 5 and 6, all chromosomes of population will be applied crossover and mutation depending on model's crossover rate and mutation rate. Then, in step 7, evaluation for fitness is similar as step 3. In step 8, population will be updated according fitness of each chromosome. Namely, if chromosome has high fitness, it will occur more times than low-fitness chromosome after selecting. Then process will back to step 4, and step 4 will continue to step 5, until termination is satisfied. If accuracy curve for generations is convergent, termination is satisfied.

In each generation, model will record the chromosome which has highest fitness among population. After all generations finished, the fittest chromosome among highest-fitness chromosome of each generation is global fittest chromosome. The subset of features related this fittest chromosome is the most optimal features subset.

2.4 evaluation method

In terms of classification problem, confusion matrix is a kind of method for summarizing the performance of classification algorithm [7]. Meanings of each element in matrix could be described as below:

	Predicted:0	Predicted:1
Actual:0	TN	FP
Actual:1	FN	TP

Table 1: Confusion matrix

From, Table 1, we can calculate its accuracy based on below standard:
Accuracy = (TP +TN)/ total.
Misclassification Rate: (FP +FN)/total

As classification problem, this method is explicit to measure its accuracy. TN means the number of patterns aligned to correct class 0. FP means the number of patterns which predicted label is 1, but actual label is 0. FN represents that numbers of patterns which actually belong to 1, are predicted in 0. TP represents that numbers of patterns which is predicted as 1, actually belong to 1. As for an efficient model, if evaluated by confusion matrix, its TN and TP are very high, and its FN and FP will be very low, close to 0.

3 Experiment and result

3.1 Experiment implement

In this section, I loaded dataset *Wisconsin Diagnostic Breast Cancer (WDBC)*, and preprocess input data:

(1) Data should be checked whether it contains null value or invalid value. If data is not clear, use code to format data and recognize invalid value.
(2) Shuffle data randomly, to reduce bias of data, because in training process, different initial pattern and order of pattern will influence the result.
(3) If datafile without separately test set, splitting data into two parts, one is training set, about 80% of whole dataset, another part about 20% is considered as test set. Independence between these two sets need to be guaranteed, to treat test set as a validation data.

After preprocess, I executed one class to define neural network, which basic technique has been introduced in previous method section. As for structure about genetic algorithm, model has following part: getting fitness function, selection function, crossover function, mutation function, training function, test function. In main part, all functions are called to finish genetic algorithm. Additionally, population initialization is random generation, rather than repeat one random chromosome, because it is easier to convergence. In training function, Bimodal Distribution Removal algorithm was added to plot accuracy curve.
As for parameter, parameter has three parts:

(1) Parameter of neural network: according to the most common rule of thumb, slight decreasing principle should be applied. In any layer, numbers of neurons should be in range from one to the number of input minus the number of output. Thus, in two hidden layers, numbers of neurons are 15 and 7. Learning rate is 0.001, and number of epochs is 200.
(2) Parameter of Genetic Algorithm: size of population is 100. Size of chromosome is 30, because features size is 30. Rate of crossover and mutation is all 0.01. Number of generation is 50.
(3) Parameter of BDR: alpha is 1.0.

3.2 Result and discussion

Firstly, I compared performance of BDR algorithm and normal neural network in training process:

```
Epoch [9301/10000] Loss: 0.3804  Accuracy: 92.43 %
Epoch [9401/10000] Loss: 0.3745  Accuracy: 92.43 %
Epoch [9501/10000] Loss: 0.3687  Accuracy: 92.43 %
Epoch [9601/10000] Loss: 0.3630  Accuracy: 92.65 %
Epoch [9701/10000] Loss: 0.3574  Accuracy: 93.10 %
Epoch [9801/10000] Loss: 0.3519  Accuracy: 93.32 %
Epoch [9901/10000] Loss: 0.3464  Accuracy: 93.32 %
```

Figure 2: Without BDR, loss change in different epoch

```
Epoch [2001/3000] Loss: 0.2900  Accuracy: 97.36 %
Epoch [2101/3000] Loss: 0.2737  Accuracy: 98.32 %
Epoch [2201/3000] Loss: 0.2649  Accuracy: 99.04 %
Epoch [2301/3000] Loss: 0.2573  Accuracy: 99.04 %
Epoch [2401/3000] Loss: 0.2744  Accuracy: 96.15 %
Epoch [2501/3000] Loss: 0.2296  Accuracy: 100.00 %
Epoch [2601/3000] Loss: 0.2225  Accuracy: 100.00 %
```

Figure 3: Without BDR, loss change

Figure 2 and Figure 3 shows that if adding BDR algorithm in training process, loss curve of model will be convergent earlier than normal neural network without BDR. In terms of classification problem, confusion matrix for test set could explain accuracy of classifier.

	Predicted:0	Predicted:1
Actual:0	50	0
Actual:1	1	69

Table 2: Confusion matrix for normal neural network

	Predicted:0	Predicted:1
Actual:0	0	37
Actual:1	0	73

Table 3: Confusion matrix for applying BDR

From Table 2 and Table 3, we can see that even if learning curve can be convergent earlier based on BDR, however, accuracy in test set will be lower than normal neural network. Accuracy of normal network is 99.17%. Accuracy of BDR model is 66.36%.

According to result of pattern removal algorithm, we can see that this algorithm could accelerate the process of training. Training process will be terminated in earlier epochs than original network. Namely, the accuracy in training set will up to almost 100% more quick than original. Thus, pattern removal algorithm indeed reduces the number of epochs which training process need. However, Test set cannot be adjusted. If test set also have outliers, the accuracy in test set would be influenced by outliers. However, sometimes removing patterns to reduce variance will enlarge its bias. As a result, after training process, model's accuracy in test set would reduce because test set as a real dataset will contain outliers. In short, whether removing patterns in training set should depends on actual situation.

Then I adopted genetic algorithm to show results. After 50 generation, I recorded the fittest chromosome in each generation, to get global optimization, because curve is not convergent. I plotted its fitness curves:

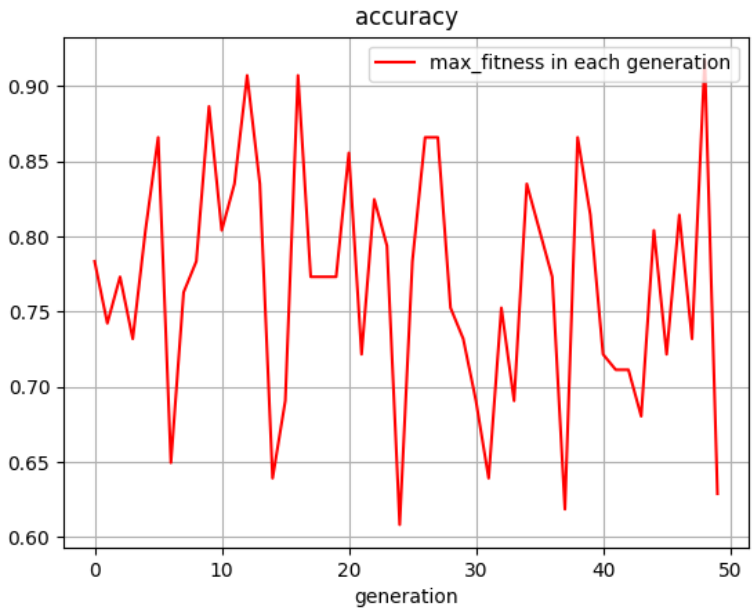

Figure 4: adding BDR in training process and applying GA in 50 generation,

plotting the highest fitness in each generation

Figure 4 shows that after some generation, the fittest individual could have higher fitness than previous generation. In last generation, accuracy based on subset of features could achieve about 0.95. Additionally, the size of global fittest feature subset just be half of size of original features. Thus, model can achieve high accuracy just relying on subset of features.

As for training part, because I adopted Bimodal distribution removal algorithm, model will be convergence earlier than normal neural network in training process. To reduce its complexity, I just trained model 500 epochs for each subset of features. As a result, BDR will begin every 20 epochs after 200 epochs. From result, we can see that accuracy in test set will be improved

by feature selection relying on genetic algorithm. Then, I applied genetic algorithm to do feature selection again without BDR, the result shows:

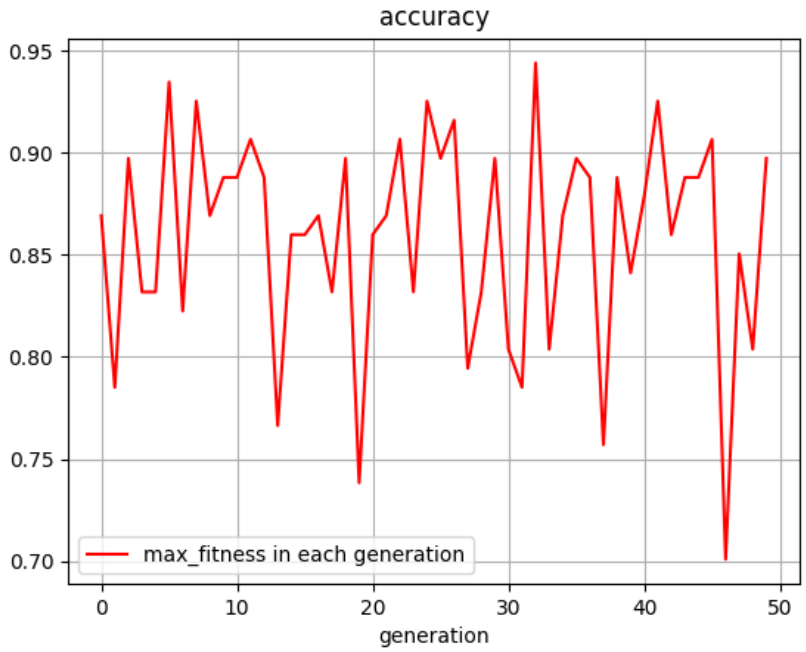

Figure 5: without BDR in training process and applying GA in 50 generation, plotting the highest fitness in each generation

From Figure 5, we can see that accuracy curve is similar as previous curve, which applying BDR in training function. Combining above results, we can point out that feature selection by genetic algorithm could reduce the negative influence of BDR in some cases. Feature selection indeed has following advantages:

(1) Genetic algorithm could accelerate the training process. According to training process, training epoch of each subset of features just need 500 to get excellent performance in test process.
(2) As for this dataset, it has 30 features and only 699 patterns. Applying BDR model will delete outliers to reduce number of patterns. Feature selection by genetic algorithm could avoid overfitting to enhance its performance in test set. Thus, it could improve generalization of BDR model.
(3) It reduces the dimensions of model to solve classification, to reduce the computational complexity.

If compared with paper related this dataset, some researchers also design classifier based on this dataset, I will compare their results with my own result. Researchers shows 97.38% as the best accuracy of his model [3], and researchers pointed out that his prospective accuracy lies in 95.5%-98.5 [2]. From confusion matrix, we can see that testing accuracy of my normal model is 99.17%, sometimes the accuracy might be almost 100%. Adding BDR and GA, model could achieve 93%- 98%, just depending on subset of features. It shows that after training process, my neural network is an efficient classifier, because it just need subset of features, less complexity, rather than all features.

4 Conclusion and future work

In this article, I compared the Bimodal Distribution Removal with the normal artificial neural network relying on its confusion matrix and accuracy, to figure out its advantages and limitation. Additionally, I applied the genetic algorithm to do feature selection to improve the performance of BDR model. According to experiment result, only optimizing and accelerating the training process, this method also improves the generalization of BDR. Additionally, reduction of dimensions is an effective way to avoid overfitting.

According to the conclusion, the algorithm in this article still has some limitations. Firstly, genetic algorithm will cost so long time to convergence. Secondly, this algorithm can't guarantee a global solution. As for each subset of features, the dataset will be trained in same neural network under same number of epochs. This implement ignores this situation that one subset

could have better performance if it is trained in different cases. Based on above limitation, a model could be improved by following method, improvement in initialization population, rather than randomly initialization, and recording loss change in the training process for each feature subset.

References

1. Blake C, Merz C (1998) UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set. In: Archive.ics.uci.edu. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic). Accessed 30 May 2018

2. Wolberg W, Street W, Mangasarian O (1994) Machine learning techniques to diagnose breast cancer from image-processed nuclear features of fine needle aspirates. Cancer Letters 77:163-171. doi: 10.1016/0304-3835(94)90099-x

3. Ratanamahatana C, Gunopulos D (2003) Feature selection for the naive bayesian classifier using decision trees. Applied Artificial Intelligence 17:475-487. doi: 10.1080/713827175

4. Slade P, Gedeon T (1993, June) Bimodal distribution removal. In International Workshop on Artificial Neural Networks 249-254. Springer, Berlin, Heidelberg.

5. Guyon I, Elisseeff A (2003,March) An Introduction to Variable and Feature Selection. In: Jmlr.csail.mit.edu. http://jmlr.csail.mit.edu/papers/v3/guyon03a.html. Accessed 30 May 2018

6. Gedeon T (2018) Genetic Algorithms for Feature Selection.

7. Brownlee J (2016) What is a Confusion Matrix in Machine Learning. In: Machine Learning Mastery. https://machinelearningmastery.com/confusion-matrix-machine-learning/. Accessed 30 May 2018

8. Yang J, Honavar V (1998) Feature Subset Selection Using a Genetic Algorithm. Feature Extraction, Construction and Selection 117-136. doi: 10.1007/978-1-4615-5725-8_8