UNIVERSITY OF CAMBRIDGE
Cambridge Service Alliance

Classification of Noisy Data: An Approach Based on Genetic Algorithms and Voronoi Tessellations

Abdul Rauf Khan, Henrik Schiøler, Torben Knudsen, Murat Kulahci and Mohamed Zaki

This is a working paper.

Why this paper might be of interest to Alliance Partners:

In today's manufacturing paradigm predictive capabilities in manufacturing can be considered a strategic advantage over competitors. Due to the technological advancement and globalization of the world's economy, manufacturing industry is going through a transitional phase at an unprecedented pace. Digitalization of manufacturing, the Industrial Internet of Things (IIoT) and the fourth industrial revolution are concepts that are getting monumental attention. The advantageous feature of these concepts is not only the idea of machine to machine (M2M) communication, but as a step forward, the deployment of smart machines in manufacturing i.e., machines that are capable of making more informed and automated decisions. The key to manifesting this idea of informed and automated decision making is an intelligent handling and astute analysis of sensor data. In this paper we are presenting a classification methodology especially designed to deal with the issues related to the sensor data analysis, and failures classification in manufacturing.

November 2016

Find out more about the Cambridge Service Alliance:
Linkedin Group: Cambridge Service Alliance
www.cambridgeservicealliance.org

UNIVERSITY OF CAMBRIDGE
Cambridge Service Alliance

Classification of Noisy Data: An Approach Based on Genetic Algorithms and Voronoi Tessellations

Abdul Rauf Khan,[1] Henrik Schiøler,[1] Torben Knudsen,[1]
Murat Kulahci[2] and Mohamed Zaki[3]

[1]Department of Electronic Systems, Aalborg University, Denmark
[2]Department of Applied Mathematics and Computer science, Technical University of Denmark and Lulea University of Technology, Sweden
[3]Institute for Manufacturing, Department of Engineering, University of Cambridge, UK

Classification is one of the major constituents of the data-mining toolkit. The well-known methods for classification are built on either the principle of logic or statistical/mathematical reasoning for classification. In this article we propose: (1) a different strategy, which is based on the portioning of information space; and (2) use of the genetic algorithm to solve combinatorial problems for classification. In particular, we will implement our methodology to solve complex classification problems and compare the performance of our classifier with other well-known methods (SVM, KNN, and ANN). The results of this study suggest that our proposed methodology is specialized to deal with the classification problem of highly imbalanced classes with significant overlap.

Classification in data mining is a supervised analogy for clustering in an unsupervised case, where the differentiation factor is the basic information available to extract insights from the data. In the case of supervised learning, additional information about the class labels transmute learning problems into the selection of suitable separation boundaries for already known classes, whereas in the unsupervised case the objective is often to identify the class labels [10]. With this brief introduction to the working domain, this article is particularly concerned with the classification challenges of imbalanced, as well as noisy, data; in other words, a significant overlap between the classes.

As a result of the wide application and availability of increasing amounts of information, there exists a relatively developed toolkit for encountering issues related to classification problems. The well-recognized classification methods can be grouped into two broad categories; logic-based algorithms and statistical learning algorithms [9]. Classification trees and decision trees are prominent among logic-based algorithms. The classification regression trees (CART) [1] [12] [18], ID3 [19], C4.5 [20], SLIQ [13] and PUBLIC Algorithm [21] are some of the popular tree-based algorithms. Most of these algorithms are designed to solve specialized problems; for example, ID3 is limited to discrete attributes, whereas C4.5 does not have this restriction and SLIQ, in particular, addresses scalability and flexibility issues [16]. Another renowned family of methods is: perceptron-based [24] algorithms. The well-known, artificial neural network (ANN) can be seen as an extension of single-layer perception into a multi-layer perceptron [28]. The artificial neural network and the concept of deep learning are among the rapidly growing research areas in the field of machine learning. There exist a number of different network designs and architectures based on the idea of ANN. Some recent additions are the Recurrent Neural Network (RNN) and the convolutional neural network (CNN), which have attracted significant attention in the research community, as well in industry. Schmidhuber (2015) [25] provided a brief overview of the methods of deep learning and neural networks. In contrast to the logical and perceptron-based methods, the statistical/mathematical learning approaches are based on the underlying probability model [9]. Discriminant analysis and Bayes rules can be considered the distinguished members of this class. In the case of linear classification and continuous data, linear discriminant analysis (LDA) or fisher linear discriminant analysis (an extension of LDA for more than two classes) [6] are used widely in the literature, whereas in the case of categorical data, discriminant correspondence analysis [26] is relatively well known. Second, for non-linear classification problems, methods such as the naive Bayes classifier, Bayesian networks [28] and instance-based learning principles (e.g. k-nearest neighbour) [2] belong to the statistical/mathematical learning class. Moreover, the newest addition to supervised learning approaches are support vector machines (SVMs) [27], which are also based on mathematical thinking. There are a number of studies [29] [9] in the literature on the comparison between the different methodologies; in general, artificial neural networks (ANN), support vector machine (SVM) and k-nearest neighbour (KNN) are considered to be the prominent algorithms that are specialized to deal with multidimensional and continuous types of data. On the other hand, logic-based methods (classification tress) have a good reputation for dealing with discrete types of data [9]. The selection of a universally best-performing (minimum misclassification probability) classifier is a stumbling block in the field. Moreover, this choice becomes more difficult in the case of significant overlap and a high imbalance between the classes. For this reason, determination of a best classifier on the basis of overall minimum misclassification probability alone is a challenge without specifying the correct weight matrix.

The motivational problem for this work is the classification of good and bad categories in highly optimized industrial processes, where the pass class contains 99 per cent of the data, whereas the fail class only contains the remaining 1 per cent. Another important fact is the misclassification cost; in most cases the misclassification of fail into pass is costlier than the other way round (pass into fail). A high imbalance (small probability of failure), along with a significant overlap (in the pass and fail class), disrupt the whole probabilistic paradigm of weight assignment and boundary selection. In this article we propose a methodology to deal with these issues, which can be used for classification in the case of imbalanced and noisy data.

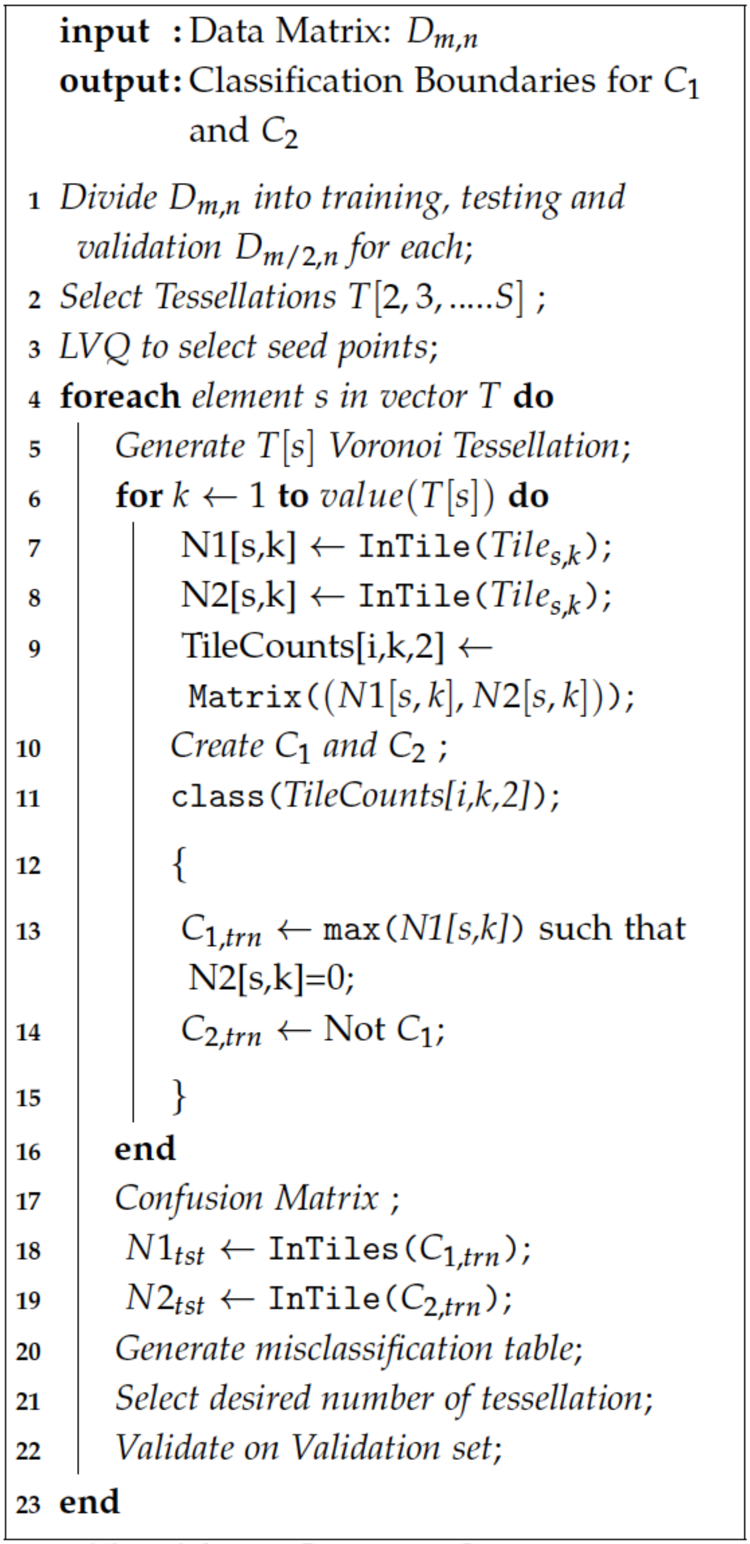
The structure of this article is as follows; we first present the methodology to classify imbalanced and noisy data; in the next section we implement our proposed methodology to the range of different problems. In the same section we present the results of comparison with other well-known methods (SVM, KNN and ANN). We conclude the article with a discussion on the learning capabilities of the proposed methodology and possible extensions of this work.

Methodology

Fundamentally, our classification strategy is built on the following three pillars; selection of seed by learning vector quantization (LVQ); Voronoi tessellation based on selected seed; and optimization through genetic algorithm. The core idea is to divide (or tessellate) the information space into a pre-selected number (of tiles) and then to group these tiles by maximizing our objective function through the genetic algorithm.

As the optimization problem becomes combinatorial in nature, the evolutionary type of algorithm is best suited to solving this problem. Our approach selects the best-represented seeds for the Voronoi tessellation by using the learning vector quantization (LVQ)[8][22] algorithm. Then the algorithms tessellate the information space by using the Voronoi tessellation [11]. As a result of its nice properties and resemblance to the k-nearest neighbour approach, the Voronoi tessellation is not new in the pattern recognition; it is already part of literature such as [15] [4]. Finally, the genetic algorithm is used to maximize objective function and, like the previous two algorithms, it is also widely used for different learning tasks such as those featuring extraction [30], the discovery of classification rules [5] and adaptive learning [3]. The novelty here is to develop a learning strategy by combining LVQ, Voronoi tessellation and genetic algorithm to achieve an optimal classification boundary focusing on the problem of significant imbalance and noise. The algorithmic view of the proposed learning strategy is presented in Algorithm 1.

The number of Voronoi tiles (or tessellations) is an important parameter in our learning strategy.



```
input  : Data Matrix: $D_{m,n}$
output : Classification Boundaries for $C_1$
         and $C_2$

Divide $D_{m,n}$ into training, testing and
  validation $D_{m/2,n}$ for each;
Select Tessellations $T[2, 3, .....S]$ ;
LVQ to select seed points;
foreach element s in vector T do
    Generate $T[s]$ Voronoi Tessellation;
    for $k \leftarrow 1$ to $value(T[s])$ do
        N1[s,k] $\leftarrow$ InTile($Tile_{s,k}$);
        N2[s,k] $\leftarrow$ InTile($Tile_{s,k}$);
        TileCounts[i,k,2] $\leftarrow$
         Matrix(($N1[s,k], N2[s,k]$));
        Create $C_1$ and $C_2$ ;
        class($TileCounts[i,k,2]$);
        {
        $C_{1,trn} \leftarrow$ max($N1[s,k]$) such that
         N2[s,k]=0;
        $C_{2,trn} \leftarrow$ Not $C_1$;
        }
    end
    Confusion Matrix ;
     $N1_{tst} \leftarrow$ InTiles($C_{1,trn}$);
     $N2_{tst} \leftarrow$ InTile($C_{2,trn}$);
    Generate misclassification table;
    Select desired number of tessellation;
    Validate on Validation set;
end
```

Algorithm 1: Learning Strategy

In order to select the optimal number of tiles for any classification problem, we used a 100-fold cross-validation strategy. We used the measure of markedness [17] instead of overall misclassification probability for the purpose of comparison, as well as a decision statistic for model selection (number of tiles). In the case of significant imbalance (99% and 1%), the optimal classifier with equal weights will always classify the whole data set into one class; in other words, it will result in no separation between the two classes. Without doubt, the optimal classifier will minimize the overall misclassification but will have an undefined markedness value. Therefore, markedness has the advantage over the overall misclassification measure in the case of a high imbalance.

Prediction outcome

		p	**n**	**total**
actual value	**p^0**	True Positive (TP)	False Negative (FN)	P^0
	n^0	False Positive (FP)	True Negative (TN)	N^0
total		P	N	

The markedness measure is the probability that a condition (or class) is marked (or classified) by the predictor [17]. It can be defined as:

Markedness = Precision + InversePrecision - 1 (1),

where

$$\text{Precision} = \frac{\text{ActualPositives/PredictivePositives}}{\text{PredictivePositives.}}$$

By the above definition,

$$\text{Precision} = \frac{\text{TP}}{(\text{TP+FP})} \quad \text{and} \quad \text{InversePrecision} = \frac{\text{TN}}{(\text{TN+FN})}$$

By definition, markedness is the probability that the particular class is marked by a classifier; therefore, this value will lie between 0 and 1. If the value approaches 1, it refers to the good performance (precision) of the classifier, whereas a value close to 0 represents the opposite. Therefore, the straightforward selection rule for the Voronoi tiles will be the number of tiles with the maximum value for the markedness measure. An important point here is that our learning strategy is designed in such a way that we use both a training and testing set for learning, and a validation set is used to validate the knowledge. In other words, training and testing are considered a learning step.

Results

In order to test the classification capabilities of the proposed methodology, it is implemented to solve four different classification problems, as presented in Figure 1. These four test problems include different degrees of imbalance, as well as noise (or overlap), in the classes.

Problem 1 Nice separation: Problem 1, the top-left representation in Figure 1, is a well-known XOR problem with nice separation between the two classes. However, classes here are imbalanced in such a way that the counts for Class 1 and Class 2 are 9,000 and 1,000 respectively.

Problem 2 Significant overlap: In the bottom-left representation in Figure 1, an XOR problem with significant overlap is presented. Classes in Problem 2 are also imbalanced with the same numbers as in Problem 1.

Problem 3 Nice separation but highly non-linear: The top-right plot in Figure 1 is a highly non-linear classification problem, where one class is surrounded by another class. The balance between classes is the same as in the previous two problems, namely, 9,000 and 1,000 for Class 1 and Class 2 respectively.

Problem 4 Significant overlap, highly imbalanced and noisy: Problem 4 in Figure 1 represents highly imbalanced (10,000 counts for Class 1 and 200 counts for Class 2) and noisy data. However, as can be seen, there are some areas or patches (at least five) where there is a high concentration of Class 2 (red class).

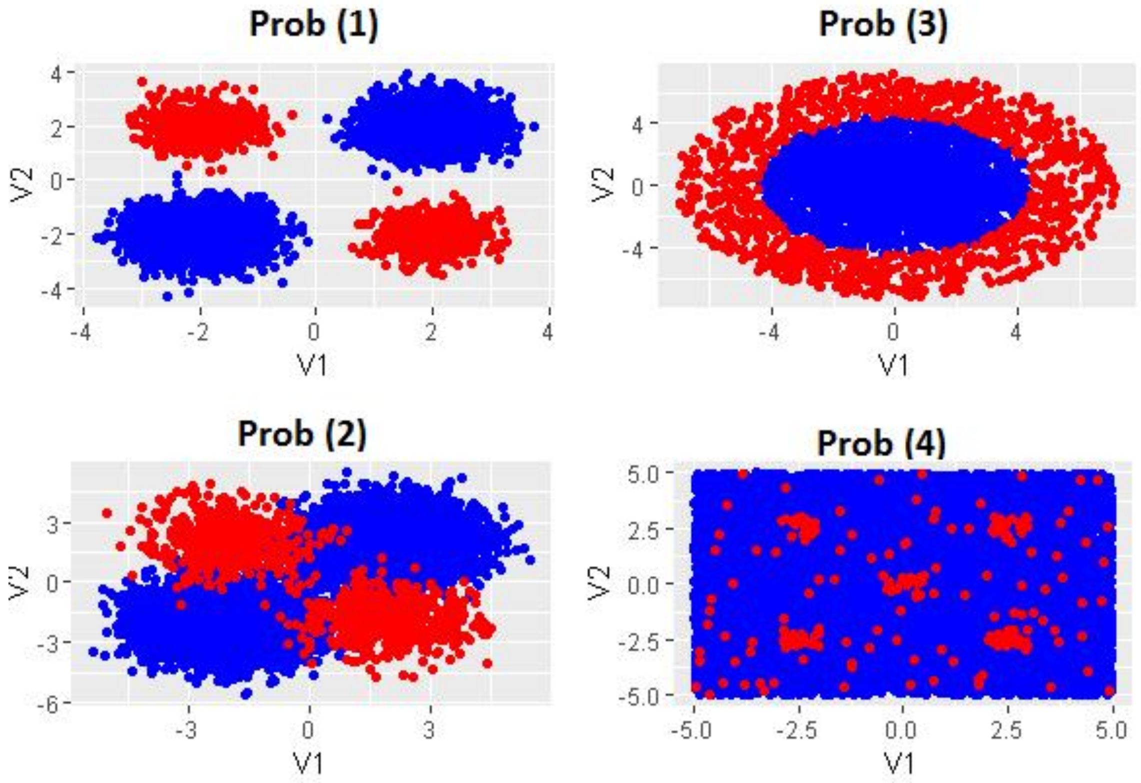


Figure 1: Classification Problems

Figure 2 represents the classification boundaries computed through the proposed methodology. The matrix of plots in Figure 2 has the same order as presented in Figure 1; in other words, the top left is a typical XOR problem with well-separated classes, and on the bottom-right corner there are significantly overlapping classes. We have observed that the methodology works fairly well in solving both relatively easy (left-hand column of Figures 1 and 2) and reasonably complex (right-hand column of Figures 1 and 2) classification problems. As discussed, because of our interest in the low-frequency class, the measure of markedness is used to select the tiles; furthermore, it is used to compare performance. We have observed that for the first three cases the markedness value is approaching 1, which is almost perfect, whereas in the last case, the frequency of one class (red dots) is significantly smaller than the second class (blue dots) (50 times), and there exists a significant noise maximum markedness score of around 0.30 (at Tiles 25).

To substantiate the need for the proposed methodology, we compared the performance of the algorithm with the existing methodologies in the literature on SVM, artificial neural network and k-nearest neighbour. We compared the results of Rr-implementation [14] [23] [22] of these well-known methods (SVM, ANN (or NNET) and KNN) with our methodology. For the purpose of comparison, we introduced F-inverse statistics along with the markedness measure (discussed earlier). F-inverse is motivated by a well-known measure in the machine-learning and data-mining literature called F1 score.

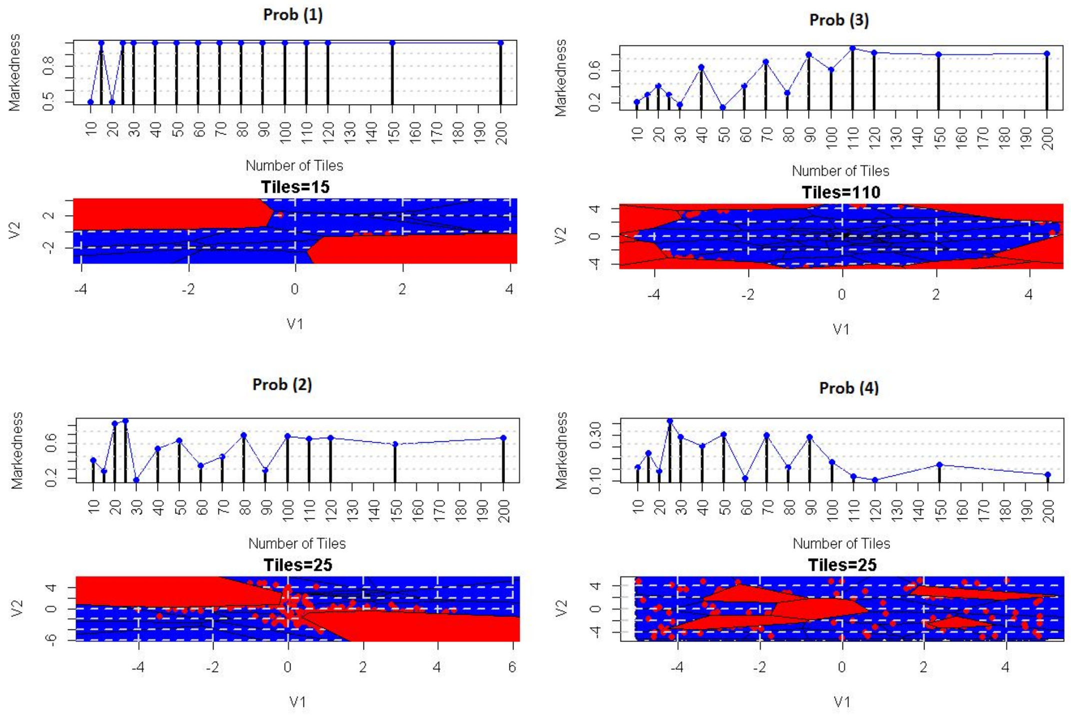


Figure 2: Classification Boundaries (test set)

$$F_1 = 2.\frac{\text{precision.recall}}{\text{precision + recall}} = \frac{2TP}{2TP + FP + FN} \quad (2)$$

F-inverse is not a widely used measure in the machine-learning and data-mining community. By definition, it is simply a transformation of F-measure, as in statistics PS+ represents a proportion of the specific agreement for the positive (+) class and PS_ represents the same the same for the negative (-) class [17]. By this definition, the value for inverse precision can be derived as follows:

$$\text{F}_{\text{1inverse}} = \frac{2TN}{2TN+FP+FN} \quad (3)$$

First, the comparison presented in Figure 3 is achieved by comparing statistics such as F-inverse and markedness. Second, the overall performance is compared with the ROC space plots presented in Figure 4. The ROC space is a two-dimensional space with false positive rate $FPR = \frac{FP}{FP+TP}$ on the x-axis and false positive rate $TPR = \frac{TP}{FP+TP}$ on the x-axis. In order to overcome the problem of testing bias, it is tested on 100 different testing sets, and the elements of the confusion matrix or misclassification probabilities are the average of 100 test sets.

Comparison with existing methods

For the purpose of comparison, R-implementation of SVM [14], ANN [23] and KNN [22] were deployed to solve the problems presented in Figure 1. In order to deal with the issue of imbalance, class weights were fixed to the class proportion. The left-hand plot in Figure 3 represents the value of the markedness measure for all four problems (on the x-axis) and all four methods (differentiated by colour). We observed that the vor_{ga} algorithm (proposed method) outperforms in the case of Prob 4, but other methods, especially KNN, performs better in the cases of Prob 1, Prob 2 and Prob 3. Moreover, the right-hand plot in Figure 3 (F-inverse score) also supports this viewpoint. In the data-mining community, comparing the performance of classifiers in the ROC space [7] is relatively common. Figure 4 represents the position of the proposed classifier along with the SVM, ANN and KNN classifiers for all four problems (see Figure 1). The black line in the plots represents the random guess and the colours differentiate the various methodologies.

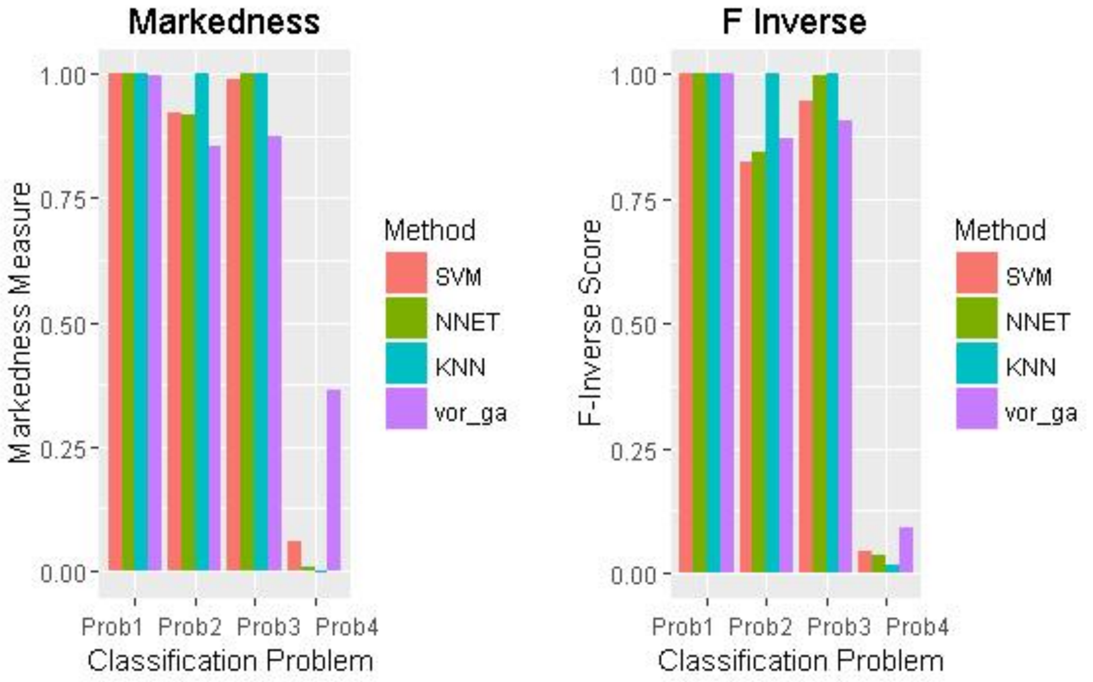


Figure 3: Comparison Between SVM, NNET, KNN and *vor-ga*

Analysis of the ROC space also supports the assertion that our proposed classifier (purple colour) performs better than other methods, especially in the case of a high level of imbalance and significant overlap (Prob 4). However, other methods outperform in the other three problems (Prob 1, Prob 2 and Prob 3). Particularly in the case of Probs 2 and 3, the purple dot is much lower than the orange (KNN), green (ANN) and blue (SVM).

Discussion

The comparison results suggest that our proposed methodology classifies reasonably well, and its performance is comparable with existing classification methodologies. Moreover, it outperforms in the case of highly imbalanced and noisy data. However, it is extremely important to mention that this comparison is only relevant to particular implementations of these well-known algorithms (SVM, NNET and KNN). In particular, in the case of neural networks there are many different architectures and learning strategies that may significantly affect performance.

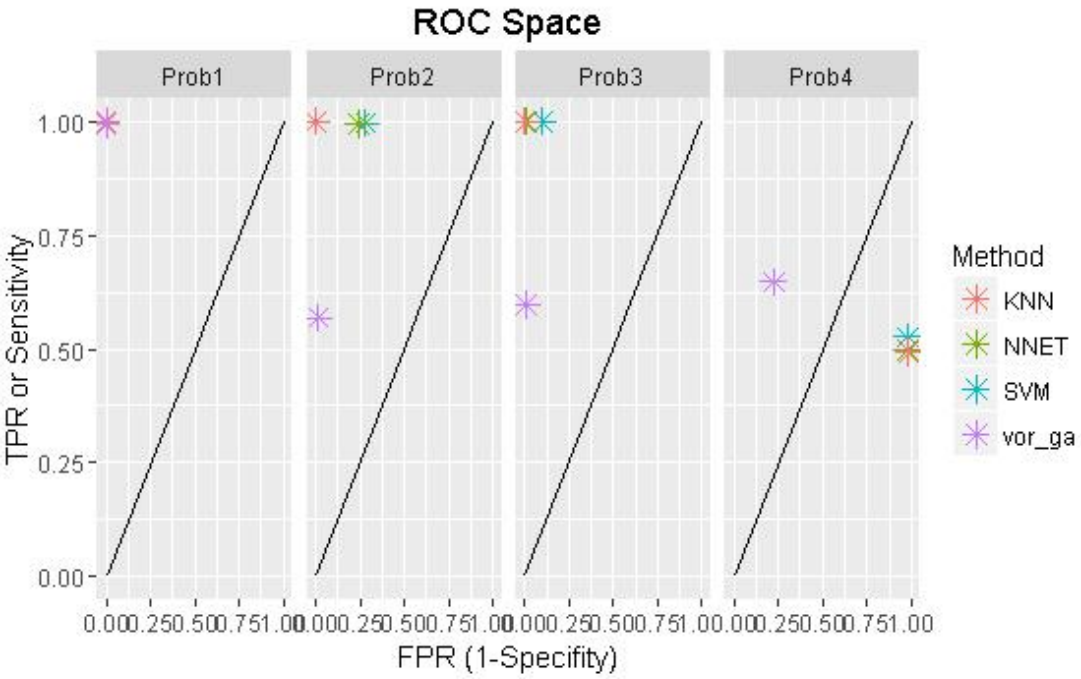


Figure 4: ROC Space

In most real-life problems, the existence of completely non-overlapping classes is simply a rare event, which is the motivation behind this work. One application of this work could be in quality and reliability analysis. In quality engineering, overlapping classes are relatively common as a result of the well-optimized processes. The probability of one class (say failure) is fairly small (for the same reason as the well-optimized processes). It is also a fact in most cases that false positives are more expensive than true negatives, but normally these weights are also unknown. *Vor$_{ga}$* solves the issue of small probabilities in such a way that our objective function for the genetic algorithm is based on real counts. On the other hand, in the case of any other probabilistic approaches, assigning best weights is itself a research question.

References

1. L Breiman, JH Friedman, R Olshen, and CJ Stone. Classification and regression trees. 1984.
2. Thomas M Cover and Peter E Hart. Nearest neighbor pattern classification. Information Theory, IEEE Transactions on, 13(1): 21–27, 1967.
3. Kenneth A DeJong and William M Spears. Learning concept classification rules using genetic algorithms. Technical report, DTIC Document, 1990.
4. Qiang Du, Vance Faber, and Max Gunzburger. Centroidal voronoi tessellations: applications and algorithms. SIAM review, 41(4): 637–676, 1999.
5. Marcos Vinicius Fidelis, Heitor S Lopes, and Alex A Freitas. Discovering comprehensible classification rules with a genetic algorithm. In Evolutionary Computation, 2000. Proceedings of the 2000 Congress on, volume 1, pages 805–810. IEEE, 2000.
6. Jerome H Friedman. Regularized discriminant analysis. Journal of the American statistical association, 84(405): 165–175, 1989.
7. James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. Radiology, 143(1): 29–36, 1982.
8. Teuvo Kohonen. The selforganizing map. Proceedings of the IEEE, 78(9): 1464–1480, 1990.
9. Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.
10. Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. Machine learning: a review of classification and combining techniques. Artificial Intelligence Review, 26(3): 159–190, 2006.
11. Der-Tsai Lee and Bruce J Schachter. Two algorithms for constructing a delaunay triangulation. International Journal of Computer & Information Sciences, 9(3): 219–242, 1980.
12. Andy Liaw and Matthew Wiener. Classification and regression by randomforest. R news, 2(3): 18–22, 2002.
13. Manish Mehta, Rakesh Agrawal, and Jorma Rissanen. Sliq: A fast scalable classifier for data mining. In Advances in Database TechnologyEDBT'96, pages 18–32. Springer, 1996.
14. David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, Friedrich Leisch, Chih-Chung Chang, Chih-Chen Lin, and Maintainer David Meyer. Package e1071. CRAN R Project, 2015.
15. Owen J Murphy. Nearest neighbor pattern classification perceptrons. Proceedings of the IEEE, 78(10): 1595–1598, 1990.
16. Xie Niuniu and Liu Yuxun. Review of decision trees. In Computer science and information technology (ICCSIT), 2010 3rd IEEE International Conference, pages 105–109, 2010.
17. David Martin Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.
18. Anantha M Prasad, Louis R Iverson, and Andy Liaw. Newer classification and regression tree techniques: bagging and random forests for ecological prediction. Ecosystems, 9(2): 181–199, 2006.
19. J. Ross Quinlan. Induction of decision trees. Machine learning, 1(1): 81–106, 1986.
20. JR Quinlan. Programs for machine learning. 1993.

21. Rajeev Rastogi and Kyuseok Shim. Public: A decision tree classifier that integrates building and pruning. In VLDB, volume 98, pages 24–27, 1998.
22. Brian Ripley and William Venables. Package class. CRAN R Project, 2015.
23. Brian Ripley and William Venables. Package nnet. CRAN R Project, 2015.
24. Frank Rosenblatt. Principles of neurodynamics. 1962.
25. Jrgen Schmidhuber. Deep learning in neural networks: An overview. Neural Networks, 61: 85–117, 2015.
26. Bernhard Scholkopft and KlausRobert Mullert. Fisher discriminant analysis with kernels. Neural networks for signal processing IX, 1(1): 1, 1999.
27. Vapnik N Vladimir and V Vapnik. The nature of statistical learning theory, 1995.
28. DE Rumelhart GE Hinton RJ Williams and GE Hinton. Learning representations by backpropagating errors. Nature, 323: 533–536, 1986.
29. Ian H Witten and Eibe Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, 2005.
30. Jihoon Yang and Vasant Honavar. Feature subset selection using a genetic algorithm. In Feature extraction, construction and selection, pages 117–136. Springer, 1998.