Data Mining Project

Abstract

Cardiac arrhythmia is a medical condition involving several characteristics that can be recognised based on different features. This report attempted to identify a classification model, by systematically mining a renowned dataset with 279 features and 452 samples of anonymised patient data. No clear winner could be found that outperformed the other algorithms. The Bayesian Network algorithm with TAN search method, the C4.5 decision tree and Random Forest all had similar results with 7% wrong assignments to the healthy class on average and an approximate overall accuracy of 80%.

Table of Contents

1 Introduction

Cardiac arrhythmia is a disease with a variety of characteristics, each of them indicated by different combinations of the patient's features. The purpose of the experiments conducted within this report is to find a model that can support physicians in making a diagnosis and, thus, choosing the best treatment for the patient.

At the beginning, the report will explain the application domain of this problem and give more details about the purpose of the research. This is followed by a description of the dataset itself and a detailed explanation of how the experiments were organised to mine it, including appropriate mining schemes, pre-processing and feature selection methods, parameter tuning, the performance metric to identify the best model, boosting techniques, and also an overview about the experimental setup. The next chapter presents the results of the experiments, compares the schemes to each other and to work that has already been done for the dataset.

2 Background

This chapter will give an overview about the application domain and explain the purpose of the research.

2.1 Application Domain

Based on the objective of the algorithm, Data Mining techniques can be categorised either as predictive or descriptive (Gorunescu, 2011). Predictive methods are used to forecast unknown values based on existing variables, like in classification or regression problems. Descriptive methods, on the other hand, discover new groups (e.g., clustering) or unexpected relationships (e.g., association rules) in the data.

The dataset discussed in this report belongs to the group of classification problems, because the overall purpose is to assign labels to samples, namely the type of arrhythmia to patients. This allocation is based on the attributes of each sample, which are called predictors in this context. Furthermore, the classes are discrete (on the contrary to regression, where the output is nominal), and the order of the samples does not influence the outcome.

In addition, each sample has already been assigned to a class, which allows researchers to verify the performance of the model. Thus, the data can be used to both train and test the model.

2.2 Purpose of Research

Cardiac arrhythmia is a medical condition consolidating irregular heartbeats, myocardial infarctions, and other disorders of the cardiac rhythm (Kaye, Furniss, & Lemery, 2010). Most forms of arrhythmia do not require treatment because they are usually not life-threatening (Kaye et al., 2010). However, it is important to detect the arrhythmia, because other diseases the patient might have can influence the risk of (sometimes severe) complications. Therefore, the most important factor to measure the performance of the applied Data Mining methods is the inverted precision rate of the first class, representing all those patients that were classified as healthy although they do suffer from arrhythmia. Secondly, as the model will be used by physicians when examining the patient, it must provide fast results, which requires a short run-time.

3 Experimental Study

This chapter clarifies at length how the research was conducted. At first, the dataset will be described, followed by a detailed explanation of how the selected mining algorithms work. After that, the modifications of the algorithm, such as pre-processing, feature selection, and parameter tuning, will be pointed out. Next, the metrics to measure the performance of the adjusted algorithms will be described, together with the used boosting techniques. Finally, an overview over the experimental setup is given.

3.1 Data Set Description

The researcher has been presented with two files, one containing background information[1] and the other holding the actual dataset[2]. Furthermore, another file could be found on the Website of Bilkent University[3], Turkey, in the Weka ARFF format, which was used for the actual Data Mining task.

According to its donor, Prof. Halil Altay Güvenir, the target of the dataset is "to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 16 groups" (Güvenir, Acar, Demiröz, & Çekin, 1997). All patients not suffering from arrhythmia are classified into the first class. Each of the classes two to 15 stands for a different kind of arrhythmia, and class 16 holds all unclassified cases.

The dataset contains 452 instances (i.e., patient data) and 279 predictors, 206 of them being linear and the other 73 being nominal. The attributes represent general facts about the patient such as age, sex, height, weight, and heart rate, and also cover a lot of different aspects of the patients' medical state which have been measured by an electrocardiogram (ECG) in combination with the IBM-Mt. Sinai Hospital Programme (Güvenir et al., 1997).

A first exploration of the dataset reveals only a few missing values. (In fact, there is only one attribute that has a noteworthy amount of missing values.) On the other hand, the dataset is very imbalanced: 245 of the 452 instances (54.2%) belong to the first class (normal ECG); whereas no patients were diagnosed with Atrio Ventricular blocks (classes 11 to 13). The remaining 207 instances are distributed unevenly between the

[1] arrhythmia.names, downloaded from AUTonline

[2] arrhythmia.data, downloaded from AUTonline

[3] The original owners of the database are professors at that university, and added it to their repository at http://axon.cs.byu.edu/data/uci_class/

other twelve classes. When visualised, no clear edges could be found which indicates a high mixture of the classes. Another characteristic that could be noticed is that there are some attributes that only have one distinct value, with all instances being assigned to that particular value. In addition, there are several attributes where almost all instances have one value in common and only very few very. It can be assumed that these attributes do not have a high influence on the assignment to a specific class.

3.2 Mining Schemes

When selecting an appropriate Data Mining algorithm, it is important to keep in mind how the dataset is structured and what purpose the Data Mining task shall fulfil. First of all, the purpose of detecting a kind of arrhythmia based on the patient's medical state makes it a classification problem, which dismisses all clustering and association techniques. Secondly, the output is discrete, which excludes regression functions. Furthermore, using Neural Networks (which are implemented as Multi-Layer Perceptron in Weka) would take too much time because of the dataset's size. With respect to Weka, which was used to perform the Data Mining task, the remaining classification methods can be divided into four groups: Bayes' algorithms,, lazy algorithms, rule-based algorithms, and decision trees.

In the Bayes category there are two major approaches, Naïve Bayes and Bayesian Networks. Naïve Bayes requires the features to be independent from each other, which we can assume in the present dataset. (Although height, weight, age and heart rate do correlate, they do not necessarily cause each other.) Bayesian Networks are directed acyclic graphs that are constructed based on probabilities (Witten et al., 2011). Therefore, they work very well when dealing with uncertain values. They are very robust and have been used in the area of medicine before (Darwiche, 2010). However, there is still some uncertainty about in which situations they perform best (Darwiche, 2010). Both algorithms work with probabilities instead of boundaries, which is of advantage in this particular dataset because of the high mixture of the data.

This mixture, however, also leads to the rejection of rule-based algorithms, because they assume only a few attributes can predict the outcome, whereas arrhythmia has several risk factors (Kaye et al., 2010). Therefore, they can't be used with this dataset. In addition, the data mixing will also make the lazy algorithms (e.g., Nearest Neighbour or

Locally Weighted Learning) perform badly, because they require some delimitation between the attributes which is not given here.

Furthermore, decision trees can be used in this dataset. A decision tree is produced by splitting the data into smaller subsets (Witten, Frank, & Hall, 2011). The most common tree is C4.5 which is called J48 in Weka. This splitting is done based on different criteria, which in the case of C4.5 is based on information gain. It can handle both continuous and discrete values and often produces very accurate results. However, it is subject to over-fitting if the data is too noisy (Witten et al., 2011).

Another approach that works efficiently on large datasets is the Random Forest algorithm (Cutler, Cutler, & Stevens, 2012). Random Forests were first introduced by Leo Breiman (Cutler et al., 2012). At the beginning, the algorithm builds several trees (see above). When a new instance should be classified, it is run through the trees, each of them "voting" for which class the instance should be assigned to. The forest then chooses the class that got most votes. The strengths of this algorithm clearly lie in its ability to process large datasets efficiently and with high accuracy (Breiman & Cutler, n.d.). However, because it builds a relatively large model, it needs a lot of memory when doing so (Alglib, n.d.).

Lastly, the Voting Feature Intervals (VFI) algorithm should be considered for comparison purposes, because it was already applied successfully on the dataset by none less than the dataset's owners (Güvenir et al., 1997).

3.3 Pre-Processing

There are only a few missing values, and almost all of them can be found in one attribute. Therefore, it is not necessary to apply special pre-processing methods for this problem. However, the dataset is highly imbalanced, with 245 instances belonging to class 1 (healthy) and no instances belonging to the classes 11 to 13 (Atrio Ventricular Blocks). The remaining 207 instances are disproportionately distributed between the residual twelve classes. Therefore, resampling methods have to be used in order to balance the dataset more and gain better results with the Data Mining algorithms. This can be done by oversampling those classes that are underrepresented. Simple rebalancing can only be used with binary problems and is, therefore, not applicable in this case. However, in Weka there is the "Resample" filter which can be used for multiclass problems. With a “biasToUniformClass” of 1 the distribution among classes was approximated to an almost even level.

After all, when using a resampling method it is important to keep in mind that the proportions do not correspond to the original class distribution anymore. This makes it necessary to provide another dataset with correct allotment for testing. In the experiments that were conducted in this report, a copy of the original dataset was used to meet this need.

Another way of dealing with class imbalance is to use a cost-sensitive meta classifier. Weka gives its user the opportunity to define a cost matrix which will be applied to the “Cost Sensitive” classifier. A cost matrix has to represent how false positives and false negatives are weighted. In the case of arrhythmia, it is very important to detect the disease, but less important to identify a particular type. Therefore, the following cost matrix was used:

0	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Table 1: Cost Matrix for Cost Sensitive classifier

This cost matrix makes all wrong assignments to and from the first class more expensive than other false decisions, and, in addition, also penalises other false class assignments.

3.4 Feature Selection

It could be noticed that a lot of the instances have the same value for certain features. Some attributes actually only have one distinctive value to which all instances are assigned, e.g. attribute 96. These features do not have a high significance to predict the class and should, therefore, be excluded in order to decrease noise.

There are two popular methods to select the best features from the dataset. Firstly, the CFS (Correlation Feature Selection) Subset Evaluator selects those attributes that are not correlated to each other but predict the class very well (Witten et al., 2011). Several search methods could be applied to it, which had different outcomes on the dataset. Best First and Greedy Stepwise found the same 25 attributes, and Linear Forward Selection showed also the same result except for two attributes. Genetic Search, on the other hand, had different outcomes based on alternations to the parameter set. However the thorough examination of these different results would go beyond the scope of this report, so it was decided to not investigate Genetic Search further. Moreover, exhaustive search is not feasible in this dataset because of the great number of attributes.

Because of the above discussed similarities and differences between the applied search algorithms, it was decided to use the Best First search for the experiments. In Weka, all search algorithms can be applied to the dataset using the Attribute Selection filter. With this it is important to notice that the attributes also have to be removed in the copy of the dataset used for testing.

The second popular method to select features is the Info Gain algorithm. This algorithm calculates how much an attribute predicts the class outcome and ranks them accordingly (Witten et al., 2011). It found 140 attributes not having a predictive value at all, and additional 63 attributes with a very low information gain. Between the 76th and the 77th attribute there was a distinctive gap (0. 09086 and 0. 05066 respectively). Therefore, the first 76 attributes were selected for further experimentation.

3.5 Parameter Tuning

Most algorithms have parameters that can be altered in order to improve their performance. For Bayesian Networks, the estimator and the search algorithm can be changed. Weka's default values are Simple Estimator and K2, respectively. K2 is one standard search algorithm for Bayesian Networks (Witten et al., 2011), together with TAN (tree-augmented Naïve Bayes). Both algorithms have been used in the experiments.

For Naïve Bayes there are no options to modify it. VFI allows varying the bias, the decision tree J48 was used with and without pruning, and Random Forests can work with different numbers of trees to improve its accuracy.

3.6 Performance Metric

When selecting a good algorithm, the most important requirement is to have a useful underlying comparison scheme. In general, it is effective to have a look at the overall accuracy, the false positive and false negative rates, and the complexity and understandability of the model. However, the criteria to measure the performance of an algorithm on the dataset must always be chosen with respect to the dataset's actual meaning and to the purpose of the Data Mining task.

In this dataset it is most important to detect arrhythmia. Therefore, the main value for measuring an algorithm's performance is the "inverted precision rate" of the first attribute, i.e. $1 - Precision_{class\ 1}$. This value represents all those patients classified as healthy that actually do suffer from arrhythmia.

The second most important value to look at is the overall accuracy. This must be taken into account to avoid extreme cases where all patients are assigned to some form of arrhythmia, because this would lead to unnecessary treatments. It also considers other patients that have been diagnosed with the wrong type of arrhythmia, and, therefore, may not get the best treatment.

Another important aspect is the run-time of the model. As it will be used in a medical environment, the actual decision making process must be kept as short as possible for physicians to make efficient use of it.

3.7 Boosting Techniques

Bagging and boosting are methods that can be applied to algorithms in order to make them perform better. In bagging, each model has the same weight, whereas in boosting models are weighted based on their accuracy (Witten et al., 2011). In Weka, these two techniques are implemented as Bagging and AdaBoostM1. They will be used to improve the three winning algorithms further.

3.8 Experimental Setups

This chapter presents the experimental plan of the experiments that were carried out. Further information about how the results can be interpreted will be provided in the next chapter. The detailed runtime logs can be found in the appendix. Rebalanced datasets were tested on a copy of the original dataset (that was also reduced to the feature set used in the model); all other models were tested with 10-fold cross-validation.

If all possible combinations of experiments were conducted, the number would have increased vastly. Therefore, several combinations that did not seem successful were left out. In general, the following approach was used: First, the algorithm was run without any pre-processing or feature selection methods applied. Then, four experiments were run to cover each pre-processing and feature selection method. Based on the results (the best marked yellow below), a few combinations were tried which were then improved by parameter tuning and boosting methods. The best results were marked in green.

Naïve Bayes

see	Pre-processing	Feature Selection	Param.	Boost.	1 – prec.	correct
B.1	none	none	default	none	0.225	62.39%
B.2	resample	none	default	none	0.202	59.96%
B.3	cost sensitive	none	default	none	0.225	62.39%
B.4	none	Cfs Subset Eval	default	none	0.211	69.91%
B.5	none	Information Gain	default	none	0.215	67.92%
B.6	resample	Cfs Subset Eval	default	none	0.142	71.68%
B.7	resample	Cfs Subset Eval	default	Bagging	0.138	69.25%
B.8	resample	Cfs Subset Eval	default	AdaBoos	0.126	67.04%

Bayesian Network

see	Pre-processing	Feature Selection	Param.	Boost.	1 – prec.	correct
C.1	none	none	default	none	0.216	69.91%
C.2	resample	none	default	none	0.096	80.75%
C.3	cost sensitive	none	default	none	0.214	69.91%

C.4	none	Cfs Subset Eval	default	none	0.158	75.66%
C.5	none	Information Gain	default	none	0.192	70.35%
C.6	resample	none	TAN	none	0.093	81.86%
C.6	resample	none	TAN	Bagging	0.16	79.87%
C.6	resample	none	TAN	AdaBoos	0.093	81.86%

J48 Decision Tree (C4.5)

see	Pre-processing	Feature Selection	Params.	Boost.	1 – prec.	correct
D.1	none	none	default	none	0.231	64.38%
D.2	resample	none	default	none	0.103	71.02%
D.3	cost sensitive	none	default	none	0.231	64.38%
D.4	none	Cfs Subset Eval	default	none	0.199	68.36%
D.5	none	Information Gain	default	none	0.205	66.81%
D.6	resample	none	unprune	none	0.098	71.02%
D.7	resample	none	unprune	Bagging	0.053	72.57%
D.8	resample	none	unprune	AdaBoos	0.062	79.42%

Random Forest

see	Pre-processing	Feature Selection	Param.	Boost.	1 – prec.	correct
E.1	none	none	default	none	0.339	65.93%
E.2	resample	none	default	none	0.084	66.59%
E.3	cost sensitive	none	default	none	0.296	63.94%
E.4	none	Cfs Subset Eval	default	none	0.244	74.34%
E.5	none	Information Gain	default	none	0.283	71.68%
E.6	resample	Cfs Subset Eval	default	none	0.056	73.01%
E.7	resample	Cfs Subset Eval	20 trees	none	0.08	75.44%
E.8	resample	Cfs Subset Eval	20 trees	Bagging	0.066	77.21%
E.9	resample	Cfs Subset Eval	20 trees	AdaBoos	0.053	79.2%

Voting Features Interval

see	Pre-processing	Feature Selection	Param.	Boost.	1 – prec.	correct
F.1	none	none	default	none	0.18	49.56%
F.2	resample	none	default	none	0.059	65.04%
F.3	cost sensitive	none	default	none	0.625	24.78%
F.4	none	Cfs Subset Eval	default	none	0.15	47.12%
F.5	none	Information Gain	default	none	0.162	48.67%
F.6	resample	none	bias = 2	none	0.062	62.17%
F.7	resample	none	default	Bagging	0.081	55.09%
F.8	resample	none	default	AdaBoos	0.055	58.41%

4 Results

This chapter will interpret the results and compare them with previous work.

4.1 Analysis

What could be noticed in general is that the performance of all algorithms increased when the dataset was resampled, whereas no significant increase was noticed when applying the cost sensitive matrix. Furthermore, only Naïve Bayes and Random Forest performed better with attribute selection, namely Cfs Subset Evaluator, whereas the other three algorithms did not show such behaviour. The Information Gain method did not improve the performance of any algorithm.

In Naïve Bayes, J48 and Random Forest, AdaBoostM1 outperformed Bagging, which in turn outperformed the use of no boosting methods. Bayesian Networks performed at the same level with and without AdaBoostM1, and VFI performed best with no boosting methods applied. Parameter tuning improved the outcomes of Bayesian Network (TAN search algorithm instead of K2) and Random Forest (20 trees instead of 10).

4.2 Comparison of Schemes

An overall winner can not be identified. Bayesian Network, J48 and Random Forest all performed at nearly the same level. Strictly applying the performance measures, Random Forest performed best because it assigned the lowest amount of unhealthy patients to the healthy class. However, its overall precision is slightly below the other two algorithms, which in turn have a slightly higher amount of falsely assigned healthy patients. Furthermore, no significant differences in run-time could be noticed between the three best performing models.

4.3 Comparison with Previous Work

Almost all papers that were found citing the dataset proposed new methods to mine it. In the following, some of them are introduced.

The dataset was first analysed by a group of researchers from Bilkent University, Turkey. In their studies they found that their newly developed VFI algorithm outperforms the Naïve Bayes and Nearest Neighbour algorithms (Güvenir et al., 1997).

This could not be confirmed in the experiments conducted in this report, where Naïve Bayes performed better.

In another paper, different algorithms were compared trying to find a well performing method for analysing ECG data. It was found that Bayesian Artificial Neural Networks performed best (Gao, Madden, Chambers, & Lyons, 2005).

Other papers focused on feature selection (Cohen, Ruppin, & Dror, 2005), attribute selection (Pappa, Freitas, & Kaestner, 2002) or even proposed an unsupervised mining algorithm to solve the problem (Lagus, Alhoniemi, Seppä, Honkela, & Wagner, 2005). However, no satisfying algorithm could be found so far.

5 Conclusion

This report attempted to analyse a dataset containing data to identify different types of arrhythmia. Unfortunately, no satisfying result could be found. The three algorithms declared as "winners" do only work with an accuracy of 80%, which is not very good. This shows that future work in this area is inevitable.

6 References

Alglib. (n.d.). *Decision forest*. Retrieved 22 May, 2013, from http://www.alglib.net/dataanalysis/decisionforest.php

Breiman, L., & Cutler, A. (n.d.). *Random forests - classification description*. Retrieved 27 May, 2013, from http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

Cohen, S., Ruppin, E., & Dror, G. (2005). Feature selection based on the Shapley value. *In other words, 1*, 98Eqr.

Cutler, A., Cutler, D. R., & Stevens, J. (2012). Random Forests. In C. Zhang & Y. Ma (Eds.), *Ensemble Machine Learning* (pp. 157-175): Springer US. Retrieved from http://dx.doi.org/10.1007/978-1-4419-9326-7_5. doi:10.1007/978-1-4419-9326-7_5

Darwiche, A. (2010). Bayesian networks. *Commun. ACM, 53*(12), 80-90. doi:10.1145/1859204.1859227

Gao, D., Madden, M., Chambers, D., & Lyons, G. (2005). Bayesian ANN classifier for ECG arrhythmia diagnostic system: A comparison study*IEEE.* Symposium conducted at the meeting of the Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on

Gorunescu, F. (2011). *Data Mining: Concepts, models and techniques* (Vol. 12): Springerverlag Berlin Heidelberg.

Güvenir, H. A., Acar, S., Demiröz, G., & Çekin, A. (1997, 7-10 Sep 1997). A supervised machine learning algorithm for arrhythmia analysis Symposium conducted at the meeting of the Computers in Cardiology 1997 Retrieved from http://www.cs.bilkent.edu.tr/~guvenir/publications/CIC97.pdf doi:10.1109/cic.1997.647926

Kaye, G., Furniss, S., & Lemery, R. (2010). *Fast Facts: Cardiac Arrhythmias* (1 ed.). Retrieved from http://AUT.eblib.com.au/patron/FullRecord.aspx?p=744426

Lagus, K., Alhoniemi, E., Seppä, J., Honkela, A., & Wagner, P. (2005). Independent variable group analysis in learning compact representations for data.

Pappa, G. L., Freitas, A. A., & Kaestner, C. A. (2002). A multiobjective genetic algorithm for attribute selection Symposium conducted at the meeting of the Proceedings of the 4th International Conference on Recent Advances in Soft Computing (RASC-2002)

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data Mining : Practical Machine Learning Tools and Techniques* (3 ed.). Retrieved from http://AUT.eblib.com.au/patron/FullRecord.aspx?p=634862