

A Supervised Machine Learning Algorithm for Arrhythmia Analysis

HA Güvenir, B Acar, G Demiröz, A Çekin*

Bilkent University, *Başkent University, Ankara, Turkey

Abstract

A new machine learning algorithm for the diagnosis of cardiac arrhythmia from standard 12 lead ECG recordings is presented. The algorithm is called VFI5 for Voting Feature Intervals. VFI5 is a supervised and inductive learning algorithm for inducing classification knowledge from examples. The input to VFI5 is a training set of records. Each record contains clinical measurements, from ECG signals and some other information such as sex, age, and weight, along with the decision of an expert cardiologist. The knowledge representation is based on a recent technique called Feature Intervals, where a concept is represented by the projections of the training cases on each feature separately. Classification in VFI5 is based on a majority voting among the class predictions made by each feature separately. The comparison of the VFI5 algorithm indicates that it outperforms other standard algorithms such as Naive Bayesian and Nearest Neighbor classifiers.

1. Introduction

In several medical domains the machine learning algorithms were actually applied, for example, two classification algorithms are used in localization of primary tumor, prognostics of recurrence of breast cancer, diagnosis of thyroid diseases, and rheumatology [4]. Another example is the CRLS system applied to a biomedical domain [5]. This paper presents a new machine learning algorithm for another medical problem, which is the diagnosis of cardiac arrhythmia from standard 12 lead ECG recordings. The algorithm is called VFI5 for Voting Feature Intervals. The VFI5 algorithm is similar to the VFI algorithm [2], which has been applied to a dermatological diagnosis problem [1]. The input to VFI5 is a training set of records of patients. Each record contains clinical measurements, from ECG signals, such as QRS duration, RR, P-R and Q-T intervals and some other information such as sex, age, weight, together with the decision of a cardiologist. There are a total of 279 attributes (features) per patient in a record. Diagnosis of the cardiologist is either normal or one

of 15 different classes of arrhythmia. VFI5 is a supervised, inductive and non-incremental algorithm for inducing classification knowledge from examples. The knowledge representation is based on a recent technique called Feature Intervals, where a concept (class) is represented by the projections of the training cases on each feature (attribute) separately. Classification in VFI5 is based on a majority voting among the class predictions (votes) made by each feature separately. A feature makes its prediction based on the projections of training instances on that feature. The VFI5 algorithm can incorporate further information about the relevancy of a feature during the voting process. Therefore, it uses a weighted majority voting, where the weight of a feature represents its relevancy. We have also developed a genetic algorithm to learn the respective weights of features. The comparison of the VFI5 algorithm indicates that it outperforms other standard algorithms such as Naive Bayesian classifier assuming normal distribution for linear feature (NBCN) and the Nearest Neighbor (NN) classifiers. On the same dataset of ECG recordings, NBCN and NN performed with an accuracy of 50% and 53%, respectively; whereas VFI5 achieved an accuracy of 62%. The paper describes the VFI5 algorithm, and its application to diagnosis of cardiac arrhythmia. A detailed empirical comparison of VFI5 with NBC and NN on arrhythmia dataset is given.

2. Dataset

The aim is to distinguish between the presence and types of cardiac arrhythmia and to classify it in one of the 16 groups. Currently, there are 452 patient records which are described by 279 feature values. Class 01 refers to *normal ECG*, class 02 to *Ischemic changes (Coronary Artery Disease)*, class 03 to *Old Anterior Myocardial Infarction*, class 04 to *Old Inferior Myocardial Infarction*, class 05 to *Sinus tachycardy*, class 06 to *Sinus bradycardy*, class 07 to *Ventricular Premature Contraction (PVC)*, class 08 to *Supraventricular Premature Contraction (PVC)*, class 09 to *Left bundle branch block*, class 10 to *Right bundle branch block*.

class 11 to 1. degree AtrioVentricular block, class 12 to 2. degree AtrioVentricular block, class 13 to 3. degree AtrioVentricular block, class 14 to Left ventricule hypertrophy, class 15 to Atrial Fibrillation or Flutter, and class 16 refers to the rest. The first 9 features are f_1 : Age; f_2 : Sex; f_3 : Height; f_4 : Weight; f_5 : the average QRS duration in msec.; f_6 : the average duration between onset of P and Q waves in msec.; f_7 : the average duration between onset of Q and offset of T waves in msec.; f_8 : the average duration between two consecutive T waves in msec.; f_9 : the average duration between two consecutive P waves in msec. The features from f_{10} to f_{14} are the vector angles in degrees on front plane of QRS (f_{10}), T (f_{11}), P (f_{12}), $QRST$ (f_{13}), and J (f_{14}) respectively. The feature f_{15} is *heart rate* which is the number of heart beats per minute. The following 11 features are measured from the **DI** channel; f_{16} : average width of Q wave in msec.; f_{17} : average width of R wave in msec.; f_{18} : average width of S wave in msec.; f_{19} : average width of R' wave in msec.; f_{20} : average width of S' wave in msec.; f_{21} : number of intrinsic deflections; f_{22} : existence of diphasic R wave (boolean); f_{23} : existence of notched R wave (boolean); f_{24} : existence of notched P wave (boolean); f_{25} : existence of diphasic P wave (boolean); f_{26} : existence of notched T wave (boolean); f_{27} : existence of diphasic T wave (boolean). The above 11 features measured for the **DI** channel are all measured for the **DII** (features f_{28} - f_{39}), **DIII** (features f_{40} - f_{51}), **AVR** (features f_{52} - f_{63}), **AVL** (features f_{64} - f_{75}), **AVF** (features f_{76} - f_{87}), **V1** (features f_{88} - f_{99}), **V2** (features f_{100} - f_{111}), **V3** (features f_{112} - f_{123}), **V4** (features f_{124} - f_{135}), **V5** (features f_{136} - f_{147}), and **V6** (features f_{148} - f_{159}) channels. The following 9 features are measured from the **DI** channel: *J point depression* (f_{160}) measured in milivolts, *amplitude of Q wave* (f_{161}) measured in milivolts, *amplitude of R wave* (f_{162}) measured in milivolts, *amplitude of S wave* (f_{163}) measured in milivolts, *amplitude of R' wave* (f_{164}) measured in milivolts, *amplitude of S' wave* (f_{165}) measured in milivolts, *amplitude of P wave* (f_{166}) measured in milivolts, *amplitude of T wave* (f_{167}) measured in milivolts, $QRSA$ (f_{168}) which is the sum of the areas of all segments divided by 10, $QRSTA$ (f_{169}) which is equal to $QRSA + 0.5 \times \text{width of T wave} \times 0.1 \times \text{height of T wave}$. The above 9 features measured for the **DI** channel are all measured for the **DII** (features f_{170} - f_{179}), **DIII** (features f_{180} - f_{189}), **AVR** (features f_{190} - f_{199}), **AVL** (features f_{200} - f_{209}), **AVF** (features f_{210} - f_{219}), **V1** (features f_{220} - f_{229}), **V2** (features f_{230} - f_{239}), **V3** (features f_{240} - f_{249}), **V4** (features f_{250} - f_{259}), **V5** (features f_{260} - f_{269}), and **V6** (features f_{270} - f_{279}) channels. The values of these features have been measured using the IBM-Mt. Sinai Hospital program.

About 0.33% of the feature values in the dataset are missing. Class distribution of this dataset is very unfair and instances of classes 11, 12, and 13 do not exist in the current dataset. Class 01 (normal) is the most frequent one. Although the ECG of some patients show the characteristics of more than one arrhythmia, in constructing the dataset it is assumed that no patient has more than one cardiac arrhythmia.

3. The VFI5 Algorithm

The VFI5 classification algorithm is a feature projection based algorithm. The feature projection based concept representation has started with the work by Güvenir and Şirin [3]. The VFI5 algorithm represents the concept with intervals separately on each feature, and makes a classification based on feature votes. It is a non-incremental classification algorithm; that is, all training examples are processed at once. Each training example is represented as a vector of either nominal (discrete) or linear (continuous) feature values plus a label that represents the class of the example. From the training examples, the VFI5 algorithm constructs intervals for each feature. An interval is either a *range* or *point* interval. A range interval is defined on a set of consecutive values of a given feature whereas a point interval is defined a single set of values. For point intervals, only a single value is used to define that interval. For range intervals, on the other hand, since all range intervals on a feature dimension are linearly ordered, it suffices to maintain only the lower bound for the range of values. For each interval, a value and the votes of each class in that interval are maintained. Thus, an interval may represent several classes by storing the vote for each class.

The training process in the VFI5 algorithm is given in Figure 1. First, the *end points* for each class c on each feature dimension f are found. End points of a given class c are the lowest and highest values on a linear feature dimension f at which some instances of class c are observed. On the other hand, end points on a nominal feature dimension f of a given class c are all distinct values of f at which some instances of class c are observed. The end points of each feature f is kept in an array $EndPoints[f]$. There are $2k$ end points for each linear feature, where k is the number of classes. Then, for linear features the list of end-points on each feature dimension is sorted. If the feature is a linear feature, then point intervals from each distinct end point and range intervals between a pair of distinct end points excluding the end points are constructed. If the feature is a nominal feature, each distinct end point constitutes a point interval.

```

train(TrainingSet):
begin
  for each feature  $f$ 
  for each class  $c$ 
     $EndPoints[f] = EndPoints[f] \cup$ 
       $find\_end\_points(TrainingSet, f, c);$ 
  sort( $EndPoints[f]$ );

  if  $f$  is linear
    for each end point  $p$  in  $EndPoints[f]$ 
      form a point interval from end point  $p$ 
      form a range interval between  $p$  and
        the next endpoint  $\neq p$ 
  else /*  $f$  is nominal */
    each distinct point in  $EndPoints[f]$  forms a point interval

  for each interval  $i$  on feature dimension  $f$ 
  for each class  $c$ 
     $interval\_count[f, i, c] = 0$ 
  count_instances( $f, TrainingSet$ );
  for each interval  $i$  on feature dimension  $f$ 
  for each class  $c$ 
     $interval\_vote[f, i, c] = \frac{interval\_count[f, i, c]}{class\_count[c]}$ 
    normalize  $interval\_vote[f, i, c]$ ;
    /* such that  $\sum_c interval\_vote[f, i, c] = 1$  */
end.

```

Figure 1: Training phase in the VFI5 Algorithm.

The number of training instances in each interval is counted and the count of class c instances in interval i of feature f is represented as $interval_count[f, i, c]$ in Figure 1. These counts for each class c in each interval i on feature dimension f are computed by the *count_instances* procedure. For each training example, the interval i in which the value for feature f of that training example e (e_f) falls is searched. If interval i is a point interval and e_f is equal to the lower bound (same as the upper bound for a point interval), the count of the class of that instance (e_c) in interval i is incremented by 1. If interval i is a range interval and e_f is equal to the lower bound of i (falls on the lower bound), then the count of class e_c in both interval i and $(i - 1)$ are incremented by 0.5. But if e_f falls into interval i instead of falling on the lower bound, the count of class e_c in that interval is incremented by 1 normally. There is no need to consider the upper bounds as another case, because if e_f falls on the upper bound of an interval i , then e_f is the lower bound of interval $i + 1$. Since all the intervals for a nominal feature are point intervals, the effect of *count_instances* is to count the number of instances having a particular value for nominal feature f .

To eliminate the effect of different class distributions, the count of instances of class c in interval i of feature f is then normalized by $class_count[c]$, which is the total number of instances of class c .

The classification in the VFI5 algorithm is given in Figure 2. The process starts by initializing the votes

```

classify( $e$ ): /*  $e$ : example to be classified */
begin
  for each class  $c$ 
     $vote[c] = 0$ 

  for each feature  $f$ 
  for each class  $c$ 
     $feature\_vote[f, c] = 0$  /* vote of feature  $f$  for class  $c$  */

  if  $e_f$  value is known
     $i = find\_interval(f, e_f)$ 

  for each class  $c$ 
     $feature\_vote[f, c] = interval\_vote[f, i, c]$ 

  for each class  $c$ 
     $vote[c] = vote[c] + feature\_vote[f, c]$ ;

  return class  $c$  with highest  $vote[c]$ ;
end.

```

Figure 2: Classification in the VFI5 Algorithm.

of each class to zero. The classification operation includes a separate preclassification step on each feature. The preclassification of feature f involves a search for the interval on feature dimension f into which e_f falls, where e_f is the value test example e for feature f . If that value is unknown (missing), that feature does not participate in the classification process. Hence, the features containing missing values are simply ignored. Ignoring the feature about which nothing is known is a very natural and plausible approach.

If the value for feature f of example e is known, the interval i into which e_f falls is found. That interval may contain training examples of several classes. The classes in an interval are represented by their votes in that interval. For each class c , feature f gives a vote equal to $interval_vote[f, i, c]$, which is vote of class c given by interval i on feature dimension f . If e_f falls on the boundary of two range intervals, then the votes are taken from the point interval constructed at that boundary point. The individual vote of feature f for class c , $feature_vote[f, c]$, is then normalized to have the sum of votes of feature f equal to 1. Hence, the vote of feature f is a real-valued vote less than or equal to 1. Each feature f collects its votes in an individual vote vector $\langle vote_{f,1}, \dots, vote_{f,k} \rangle$, where $vote_{f,c}$ is the individual vote of feature f for class c and k is the number of classes. After every feature completes their preclassification process, the individual vote vectors are summed up to get a total vote vector $\langle vote_1, \dots, vote_k \rangle$. Finally, the class with the highest vote from the total vote vector is predicted to be the class of the test instance.

4. Experimental Results

For supervised concept learning (classification) tasks, the classification accuracy of the classifier is one measure of performance. The most commonly used metric for classification accuracy is the percentage of correctly classified test instances over all test instances. To measure the classification accuracy, 10-fold cross-validation technique is used in the experiments. That is, the whole dataset is partitioned into 10 subsets. The 9 of the subsets is used as the training set, and the tenth is used as the test set. This process is repeated 10 times once for each subset being the test set. Classification is the average of these 10 runs. This technique ensures that the training and test sets are disjoint. The VFI5 algorithm achieved 62% accuracy on the arrhythmia dataset.

The VFI5 learning algorithm can incorporate feature weights, provided externally, into classification. We used a genetic algorithm to learn weights of features. Using these weights, the VFI5 algorithm has achieved 68% accuracy, in the same experiments.

We have also applied some other well-known classification algorithms to our arrhythmia domain in order to compare the performance of the VFI5 classifier with them. The Naive Bayesian Classifier (NBCN), which assumes that the linear feature values of each class are normally distributed, has achieved a classification accuracy of 50% measured by 10-fold cross-validation. The classification accuracy of the classical Nearest Neighbor (NN) algorithm is 53%. Thus, the VFI5 algorithm performs better than these two other algorithms on the arrhythmia domain.

5. Conclusions

In this paper, a new supervised inductive learning algorithm called VFI5 is developed and applied to the problem of distinguishing between the presence and types of cardiac arrhythmia. The dataset is a set of patients described by a set of attributes and classified by our medical expert. The VFI5 classifier learns the concept from these preclassified examples and classifies new patients. The classification accuracy of VFI5 is higher than those of the common NBCN and NN classifiers.

Since the features are considered separately both in learning and classification, the VFI5 algorithm, in particular, is applicable to concepts where each feature, independent of other features, can be used in the classification of the concept. This separate consideration also provides a simple and natural way of handling unknown feature values. In other classification algorithms, such

as the NN algorithm, a value must be replaced by the unknown value.

Another advantage of the VFI5 classifier is that, instead of a categorical classification, it can return a probability distribution over all classes, that is a more general probabilistic classification.

The classification output of VFI5 is also comprehensible to the users via a user interface, from which the user can get more information such as the confidence of the classification, the next probable class, and whether and how much the attributes of the domain supports the final classification as well as the predicted class.

References

- [1] Demiröz G., Güvenir H. A., İlter N. Differential Diagnosis of Erythematous-Squamous Diseases using Voting Feature Intervals. In: *New Trends in Artificial Intelligence and Neural Networks*. Ankara: TAINN97, 1997: 190-194.
- [2] Demiröz G., Güvenir H. A. Classification by Voting Feature Intervals. In: *Proceedings of 9th European Conference on Machine Learning*. Prague: Springer-Verlag, LNAI 1224, 1997:85-92.
- [3] Güvenir H. A., Şirin İ. Classification by Feature Partitioning. *Machine Learning* 1996; 23:47-67.
- [4] Kononenko, I. Inductive and Bayesian Learning in Medical Diagnosis. *Applied Artificial Intelligence*, Vol. 7, 1996: 317-337.
- [5] Spackman A. K. Learning Categorical Decision Criteria in Biomedical Domains, In *Proceedings of the Fifth International Conference on Machine Learning*. University of Michigan, Ann Arbor, 1988.

Address for correspondence:
Bilkent University
Dept. of Computer Engr. & Info. Sci.
06533 Ankara Turkey
tel/fax: ++90-312-2664126
e-mail: guvenir@cs.bilkent.edu.tr