Application of ILP to cardiac arrhythmia characterization for chronicle recognition

R. Quiniou[1], M.-O. Cordier[1], G. Carrault[2], and F. Wang[2]

[1] IRISA, Campus de Beaulieu, 35042 Rennes Cedex FRANCE, (Quiniou, Cordier)@irisa.fr
[2] LTSI, Campus de Beaulieu, 35042 Rennes Cedex FRANCE, (Carrault, Wang)@ltsi3.univ-rennes1.fr

Abstract We propose to use ILP techniques to learn sets of temporally constrained events called chronicles that a monitoring tool will use to detect pathological situations. ICL, a system providing a declarative bias language, was used for the experiments on learning cardiac arrhythmias. We show how to obtain properties, such as compactness, robustness or readability, by varying the learning bias.

1 Introduction

In medical domains such as cardiology, intensive care units make use of more and more sophisticated monitoring tools. These tools have improved the surveillance and care of patients suffering from strong disorders. However, many false alarms are still generated and, from our point of view, these tools rely too much on signal processing algorithms. There exists a gap between the understanding level of clinicians and the information displayed by monitoring tools. To be more informative and explicative we think, as Lavrăc et al. [8], that monitoring tools must manipulate more abstract knowledge such as temporal relations between interesting events reflecting the patient's state. We have proposed in [2] to associate signal processing techniques with high-level temporal reasoning for patient monitoring. The first module processes input signals and outputs symbolic attributed events that feed a chronicle recognizer which attempts to detect specific patterns among these events. Chronicles are event patterns which state temporal constraints among a set of events.

As devising chronicles is not, in general, an easy task, we propose to use machine learning techniques in order to obtain accurate and interesting characterizations of pathological situations from examples of input signals related to disorders that may affect some patient. In the domain of coronary care units, the signals are multi-channel electrocardiograms (ECGs) and the situations to recognize are cardiac arrhythmias. As temporal relations among events are crucial as well as a specification language which can lead to informative explanations, we have chosen to use inductive logic programming (ILP). This is a major difference between Kardio [1] and our own approach. Kardio uses feature-based induction, thus, it can only learn predefined propositional structural relations. Target concepts are represented as first-order formulas in ILP. This makes the rules more

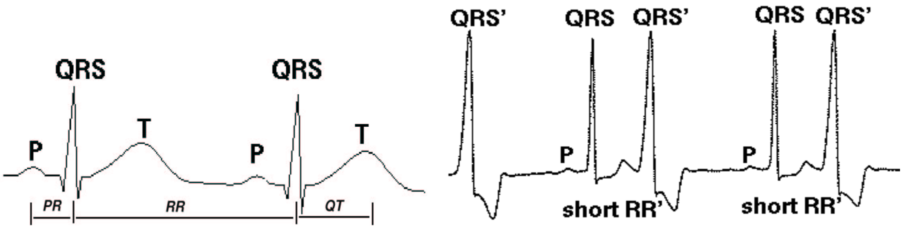


Figure1. A normal ECG (on the left) and a bigeminy ECG (on the right).

abstract and easier to understand and that is an essential point in our context as tools have to explain their results to users. Kókai et al. [6] proposed to learn attributed grammars for arrhythmia recognition in ECG from elementary curve segments. Their approach relies on grammar refinement which, they say, is not well suited to learning constraints in rules. Also, the learnt grammars specify only one cardiac cycle which is too short to describe recurrent phenomena as cardiac arrhythmias.

In this paper, our goal is to demonstrate that ILP is a powerful and smart technique that makes it relatively easy to learn knowledge adapted to the problem at hand. Precisely, we show how to play with bias specifications in order to learn concept definitions enjoying such different properties as robustness, readability or recognition efficiency. DLAB, the declarative bias language of ICL [12], has reveal quite useful and flexible to achieve this goal. The first section gives some basic knowledge about cardiac arrhythmias. The next section presents the data and learning materials. Next, we describe the results obtained on learning five arrhythmias. Finally, we conclude and give some perspectives to this work.

2 Electrocardiograms

The electrocardiogram provides very important cues for cardiac analysis and diagnosis. First of all, they can be recorded easily with non invasive leads that are put at particular locations of the body surface. Second, ECGs can be inspected visually by physicians in order to analyze the ordering and the shape of particular waves which can be related directly to the patient's heart activity. The most important waves are the P wave and the QRS complex which are related respectively to the depolarization of the atria and the depolarization of the ventricles. The ECG presents series of such waves which are organized in cardiac cycles representing a complete heart contraction and an electrical potential recovery. The normal cycle is a succession of: P wave - QRS complex - T wave. The temporal intervals between these waves are commonly used for diagnosis and noted PR, QT and RR (see figure 1, left part).

Cardiac arrhythmias are disorders of rates, rhythms and conduction originating in heart areas with dysfunctions. Arrhythmias can be recognized by specific arrangements of ECG waves satisfying temporal constraints. For example, figure 1 presents on the left a normal ECG where all heart elements (seem to) work fine. The ECG on the right is related to an arrhythmia called bigeminy, where one can note the presence of extra ventricular beats due to an ectopic focus which acts as an extra pacemaker. Bigeminy is classically defined by the wave sequence P - QRS - QRS' - P - QRS - QRS', where QRS' denotes a QRS having an abnormal shape together with the temporal constraints normal PR, short RR' and long R'R, where R' denotes the abnormal QRS. This is the kind of temporal patterns that chronicle recognition algorithms [4,3] are able to detect.

Clearly, the definition of bigeminy above is best represented by a first-order formula as it contains true relations between events. In fact, the following Prolog clause gives a straightforward specification of this definition:

$$\mathtt{bigeminy} \leftarrow \mathtt{qrs(R0,normal,P0,\_)}, \mathtt{qrs(R1,abnormal,P1,R0)}, \mathtt{rr1(R0,R1,short)}. \quad (1)$$

It states that, in bigeminy, the temporal interval between a normal and an abnormal QRS is short. To learn specifications like formula (1) we need methods that can induce temporal constraints such as simple or delayed precedence between events. Inductive logic programming (ILP) aims at inducing first-order representations of target concepts and is quite adapted to this task [10].

3 Learning algorithms and materials

In this section, we first recall some principles of ILP. Then we describe the learning data that were used to learn cardiac arrhythmias. Finally, we show how to fomulate a bias in order to improve the learning efficiency.

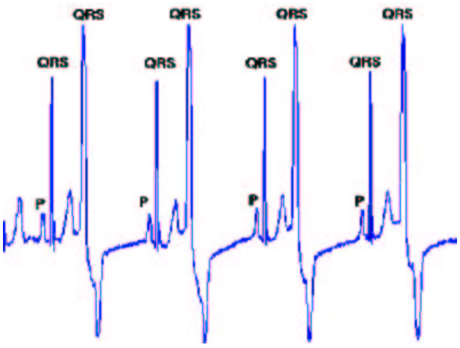
ICL: an Inductive Logic Programming system

The aim of ICL is to find a first-order theory $H \subset L_H$ that is complete (it covers all the given positive examples) and is consistent (it covers no negative examples). L_H is the hypothesis language and is generally a subset of first-order logic. An interesting feature of ILP systems is to provide the users with declarative tools which provide means to specify L_H. ICL [12] proposes a high-level concept specification language called DLAB in which the hypothesis language syntax can be defined. DLAB grammars are preprocessed in order to generate candidate hypotheses from the most general to the specific ones (under θ-subsumption).

ICL enables also multi-class learning [7]. The idea beyond multi-class learning is simple: when learning one particular class consider as positive only those examples belonging to this class and as negative all the examples belonging to the remaining classes. This is an attractive option in our case as we want to discover definitions which discriminate among several (> 2) arrhythmias.

Data

In order to assess the versatility of ICL and DLAB, we have selected a subset of arrhythmias related to different cardiac disorders involving various parts of the heart: the atria-ventricular (AV) node for the Mobitz type II arrhythmia (class mobitz2), the left bundle branch for the left bundle branch block (class lbbb) and the ventricle for bigeminy. ECGs related to a normal heart activity were also added (class normal). These 4 classes are not so difficult to separate. To augment the difficulty, we have added one class: the premature ventricular




```
begin(model(bigeminy_119_1)).
bigeminy.
wave(p1, p,   651,  normal,   null).
wave(r1, qrs, 836,  normal,   p1).
wave(r2, qrs, 1357, abnormal, r1).
wave(p2, p,   2528, normal,   r2).
wave(r3, qrs, 2686, normal,   p2).
wave(r4, qrs, 3203, abnormal, r3).
wave(p3, p,   4428, normal,   r4).
wave(r5, qrs, 4577, normal,   p3).
wave(r6, qrs, 5086, abnormal, r5).
wave(p4, p,   6279, normal,   r6).
end(model(bigeminy_119_1)).
```

Figure2. A bigeminy arrhythmia ECG and its related specification as an ICL example

contraction arrhythmia (PVC) is characterized by sparse extra contractions due to an ectopic focus. The presence of ectopic beats makes this class close to bigeminy. The fact that ectopic beats are sparse makes this class close to the normal class as large portions of PVC ECGs are normal.

Real recorded ECG examples taken from the MIT BIH database [9] were used. 20 ECGs lasting 10s each were associated to each class. Every ECG is preprocessed by a signal processing algorithm and transformed into a symbolic representation based on P and QRS events [5]. This is the same module that is used on-line to produce symbolic events that will be processed by the chronicle recognizer. It aims: *i)* at detecting and at identifying the markers of the cardiac activity, P waves, QRS complexes, *ii)* at characterizing each wave by feature vector, and *iii)* at classifying waves in normal or abnormal classes. This module is not further detailed here (see [2]) but it is of major importance as the performance of the "symbolic part" of the system relies on good input data.

Symbolic electrocardiograms

Figure 2 presents an ECG example coded as a set of prolog clauses. To each event is associated its type, its occurrence time in the ECG and a qualification (normal or abnormal) of the related wave shape. This information is coded by the predicate `wave(Event, Type, Time, Qual, Pre_event)` which states that `Event` is related to a wave of type `Type` (`p` or `qrs`), which occurred at time `Time`, the shape of which is `Qual` (`normal` or `abnormal`) and `Pre_event` just precedes `Event` on the ECG. We chose to code the structural information (order of events) as a 5th argument of the predicate `wave`. We could have used an additional relational predicate as well.

Background knowledge

The aims of background knowledge is to ease learning by bringing knowledge of the domain from which the data come from as well as search knowledge which will be used to prune the clause space. In [11], the concept of declarative learning bias is studied and its importance and properties are clearly demonstrated.

```
1      1-1:[
2        len-len:[p_wave(P1, 1-1:[normal, abnormal], R0),
3            qrs(R1, 1-1:[normal, abnormal], P1),
4            0-len:[rr1(R0, R1, 1-1:[short, normal, long]),
5              pr1(P1, R1, 1-1:[short, normal, long])]],
6        len-len:[p_wave(P1, 1-1:[normal, abnormal], R0),
7            pp1(P0, P1, 1-1:[short, normal, long])],
8        len-len:[qrs(R1, 1-1:[normal, abnormal], R0),
9            0-1:[rr1(R0, R1, 1-1:[short, normal, long])]]
10           ],
```

Figure3. Syntactic specification of a cardiac cycle in DLAB

ICL [12] comes with DLAB, a declarative language for bias specification. A DLAB grammar consists in rule templates that fixes the syntactic form of clauses defining the target concept. These templates have the form `Head <- Body` where Head and Body are DLAB terms. A term is either an atomic formula or a set specification having the form `l-h:[el1,el2,...,eln]`. Such an expression means: choose from `l` to `h` elements from the set `[el1,el2,...,eln]`. The special symbol `len` can be used to specify the total length of the list. These expressions are used as combinatorial generators that can produce all the possible instances satisfying the templates. For example, the DLAB term `p(2-len:[el1,el2,el3])` generates the following expressions:
`p(el1,el2)`, `p(el1,el3)`, `p(el2,el3)`, `p(el1,el2,el3)`.

Figure 3 shows how the specification of a cardiac cycle may be formulated in DLAB. It says that a cardiac cycle is composed of exactly one (range `1-1` line 1) of the following configurations:

- a P-wave followed by a QRS complex followed by optional (range `0-len`) temporal constraints (`pr1` and `rr1` in lines 2-5). For instance, the following expression satisfies this DLAB specification:
  `p_wave(P1, normal, R0), qrs(R1, abnormal, P1), pr1(P1, R1, long)`,
- a P-wave alone, in this case the temporal constraint between this wave and the preceding one is mandatory (lines 6 and 7),
- a QRS complex alone, in this case the temporal constraint between this wave and the preceding one is optional (lines 8 and 9).

Finally, a rule body is a sequence of such DLAB expressions telling ICL that an arrhythmia is defined by one or several cardiac cycles. Such a specification may appear quite sophisticated and restrictive. We have tried more permissive biases but either they led to prohibitive learning times or the quality of induced rules was very poor. Our objective has been to induce clauses that could be tailored in order to take into account such notions as readability, efficiency or robustness. Basing the induction on the notion of cardiac cycle enables readability since this is a concept that is commonly used by specialists for arrhythmia description or for diagnosis.

```
class(bigeminy) :-      %[13, 0, 0, 0, 0], [5, 19, 18, 18, 17]
   qrs(R0, abnormal, _), p_wave(P1, normal, R0), qrs(R1, normal, P1),
   qrs(R2, abnormal, R1), rr1(R1, R2, short).
class(bigeminy) :-      %[5, 0, 0, 0, 0], [13, 19, 18, 18, 17]
   qrs(R0, normal, _), p_wave(P1, normal, R0), qrs(R1, abnormal, P1).
class(lbbb) :-          %[0, 19, 0, 0, 0], [18, 0, 18, 18, 17]
   qrs(R0, abnormal, _), p_wave(P1, normal, R0), qrs(R1, abnormal, P1).
class(mobitz2) :-       %[0, 0, 16, 0, 0], [18, 19, 2, 18, 17]
   p_wave(P0, normal, _), equal(P0, R0),
   p_wave(P1, normal, R0), qrs(R1, normal, P1).
class(mobitz2) :-       %[0, 0, 2, 0, 0], [18, 19, 16, 18, 17]
   p_wave(P0, normal, _), equal(P0, R0),
   p_wave(P1, normal, R0), qrs(R1, abnormal, P1).
class(normal) :-        %[0, 0, 0, 17, 4], [18, 19, 18, 1, 13]
   p_wave(P0, normal, _), qrs(R0, normal, P0),
   p_wave(P1, normal, R0), qrs(R1, normal, P1),
   p_wave(P2, normal, R1), qrs(R2, normal, P2),
   p_wave(P3, normal, R2), qrs(R3, normal, P3), p_wave(P4, normal, R3).
class(pvc) :-           %[0, 0, 0, 0, 17], [18, 19, 18, 18, 0]
   p_wave(P0, normal, _), qrs(R0, normal, P0),
   p_wave(P1, normal, R0), qrs(R1, normal, P1),
   qrs(R2, abnormal, R1), rr1(R1, R2, short).
```

Figure4. Rules induced for a learning experiment on 5 classes

4 Results

The first goal of the experiments was to test whether understandable and useful arrhythmia specifications could be learnt from temporal data coming from example ECGs. A second goal was to assess the flexibility of using a declarative bias for imposing desirable properties such as readability or robustness on induced concepts. For instance, inducing the shortest clauses can be achieved by imposing only one cardiac cycle. This should bring efficiency to recognition as such rules specify less events to be recognized. Inducing longer rules enhance readability since a phenomenon regularity may be easier to assessed. A bias imposing several cycles, e.g. three or four, would be used to this purpose.

Inducing rules for five arrhythmias
Figure 4 displays the rules obtained from ICL when imposing one mandatory cardiac cycle and four optional ones. Those rules produce the shortest chronicles which are expected to enable early detection. To each rule is associated the number of examples covered by this rule in each class (respectively bigeminy, lbbb, mobitz2, normal and pvc) and the number of examples covered by its negation. For example, the list `[13,0,0,0,0]` associated to the first rule for bigeminy in figure 4 means that this rule covers 13 positive examples from class bigeminy, and none from the classes lbbb, mobitz2, normal and pvc.

Though only one cycle was mandatory, every rule states constraints on at least two cycles. Two types of temporal constraints are used: sequential con-

Table1. Learning 5 classes: statistics of 10-fold cross-validation

Set	Acc	TrueTot	FalseTot	TrAcc	Correct*Incorrect/class #
1	1.000	10	0	0.989	[1,1,5,1,2] * [0,0,0,0,0] #
2	1.000	10	0	0.989	[1,2,2,2,3] * [0,0,0,0,0] #
3	1.000	10	0	1.000	[3,2,2,1,2] * [0,0,0,0,0] #
4	1.000	10	0	1.000	[2,4,2,0,2] * [0,0,0,0,0] #
5	1.000	10	0	0.989	[2,1,2,2,3] * [0,0,0,0,0] #
6	1.000	10	0	0.989	[3,0,2,3,2] * [0,0,0,0,0] #
7	0.900	9	1	1.000	[1,2,0,5,1] * [0,0,0,1,0] #
8	1.000	10	0	0.989	[2,1,3,2,2] * [0,0,0,0,0] #
9	1.000	10	0	0.989	[2,4,0,1,3] * [0,0,0,0,0] #
10	1.000	10	0	0.989	[3,3,2,2,0] * [0,0,0,0,0] #
Tot:	9.900	99	1		

Accuracy: 0.990 (+/-0.030)   (Training set Accuracy: 0.992 (+/-0.005))

straints between events by means of the third argument of `p_wave` and `qrs` predicate literals and temporal constraints on intervals by means of predicates `pr1` and `rr1` which appear to be the most used by specialists. Two rules were necessary for mobitz2. This arrhythmia can be characterized by the episodic absence of a ventricular contraction. It is sometimes accompanied by a right bundle branch block (rbbb) provoking an enlarged QRS. This was the case for some of the examples of this class. The two rules that were obtained reflect this fact: in the first one the QRS are normal whereas in the second one the QRS are abnormal and then denote a joint rbbb.

Validation

Table 1 gives the statistics obtained after a 10-fold cross-validation on learning 5 classes. 10% of the examples were left out for test in each round. The column TrAcc gives the training accuracy and the column Acc gives the test accuracy for each round. 99.2% and 99% global accuracy was obtained for training and test respectively. These results are very good and show that accurate definitions may be induced from complex data.

The rules learnt in the previous experiments were also assessed by specialists from a qualitative point of view. Though sometimes they were surprised by some definitions which did not correspond to the general definition they were used to, they rated all the rules as being correct and relevant.

5 Conclusion

This paper has presented an application of ILP techniques to the acquisition of a set of high-level temporal patterns (or chronicles) characterizing cardiac arrhythmias. The main novelty in this application is the fact that we are dealing with temporal and structured data. The ultimate goal is to get a chronicle base which is used by a chronicle recognition tool to analyse, in an on-line monitoring context, an ECG signal and detect cardiac disorders. A description of the whole project can be found in [2]. A set of real recorded ECG signals, taken from the MIT database, has been preprocessed by a signal processing algorithm into a symbolic representation and constitute the training base.

We focus in this paper on the experimentation we did with ICL [12] and we demonstrate the interest of using a declarative bias as DLAB. According to the properties that are looked for, such as readability or robustness, different biases have been experimented and result in different sets of rules.

Two main issues are currently investigated: the first one is to cope with multiple sources of information (multichannels and multisensors). This means a new learning phase in order to get a set of chronicles able to take into account not only the temporal aspect of each signal but also the relationships existing among these different signals. The second issue concerns active cardiac devices which rely on leads located in both ventricles. These new devices can tackle both rhythmic and hemodynamic disorders but the signatures are still poorly known. We are currently experimenting our learning module on these data in order to exhibit such signatures.

References

1. I. Bratko, I. Mozetic, and N. Lavrăc. *Kardio: A Study in Deep and Qualitative Knowledge for Expert Systems*. MIT Press, 1989.
2. G. Carrault, M.-O. Cordier, R. Quiniou, M. Garreau, J.-J. Bellanger, and A. Bardou. A model-based approach for learning to identify cardiac arrhythmias. In *Proc. of AIMDM'99*, LNAI, vol. 1620. Springer Verlag, 1999.
3. C. Dousson. Crs: Chronicle recognition system. `http://crs.elibel.tm.fr/`, 2001.
4. C. Dousson, P. Gaborit, and M. Ghallab. Situation recognition: representation and algorithms. In *Proc. of 13th IJCAI*, Chambéry, France, 1993.
5. A I. Hernández, G. Carrault, F. Mora, G. Passariello, and J. M. Schleich. Multisensor fusion for atrial and ventricular activity detection in coronary care monitoring. *IEEE Trans. on Biomedical Engineering*, 1999.
6. G. Kókai, Z. Alexin, and T. Gyimóthy. Analyzing and learning ecg waveforms. In *Proc. of ILP'96*, pages 152–171, 1996.
7. W. Van Laer, L. De Raedt, and S. Dzeroski. On multi-class problems and discretization in inductive logic programming. In *Proc. of ISMIS97*, LNAI, vol.1325. Springer-Verlag, 1997.
8. N. Lavrăc, I. Kononenko, E. Keravnou, M. Kukar, and B. Zupan. Intelligent data analysis for medical diagnosis: Using machine learning and temporal abstraction. *AI Communications*, 11(3-4):191–218, 1998.
9. G.B. Moody and R.G. Mark. The MIT-BIH arrhythmia database on cd-rom and software for use with it. In *Computers in Cardiology 1990*, pages 185–188. IEEE Computer Society Press, 1990.
10. S. Muggleton. Inductive logic programming: issues, results and the challenge of learning language in logic. *Artificial Intelligence*, 114(1-2):283–296, 1999.
11. C. Nédellec, C. Rouveirol, H. Adé, F. Bergadano, and B. Tausend. Declarative bias in inductive logic programming. In L. de Raedt, editor, *Advances in Inductive Logic Programming*, pages 82–103. IOS Press, 1996.
12. L. De Raedt and W. Van Laer. Inductive constraint logic. In *Proceedings of the 5th Workshop on Algorithmic Learning Theory*, LNAI, 1995.