Towards Interval Temporal Logic Rule-Based Classification*

E. Lucena-Sánchez[1,2], E. Muñoz-Velasco[3], G. Sciavicco[1], I.E. Stan[1,4], and A. Vaccari[1]

[1]Dept. of Mathematics and Computer Science, University of Ferrara (Italy)
[2]Dept. of Physics, Informatics, and Mathematics, University of Modena and Reggio Emilia (Italy)
[3]Dept. of Applied Mathematics, University of Málaga (Spain)
[4]Dept. of Mathematical, Physical, and Computer Sciences, University of Parma (Italy)
[1]{lcnsrl,scvgdu,stnndr}@unife.it,alessandr.vaccari@student.unife.it
[3]ejmunoz@uma.es

Abstract

Supervised classification is one of the main computational tasks of modern Artificial Intelligence, and it is used to automatically extract an underlying theory from a set of already classified instances. The available learning schemata are mostly limited to static instances, in which the temporal component of the information is absent, neglected, or abstracted into atemporal data, and purely, native temporal classification is still largely unexplored. In this paper, we propose a temporal rule-based classifier based on interval temporal logic, that is able to learn a classification model for multivariate classified (abstracted) time series, and we discuss some implementation issues.

1 Introduction

Supervised classification, specifically, *supervised classification model learning* [14, 17] is one of the main tasks associated with Artificial Intelligence. Roughly, given a *data set* of *instances*, each one of which belongs to a known *class*, and each instance described by a finite set of *attributes*, supervised classification model learning is the task of learning how the values of the attributes determine the class in the context of the data set. Supervised classification models can be broadly divided into function-based, tree-based, and rule-based, depending on how the model is represented. *Function-based* classification models range from the very simple *regression* models, to *neural network* models (see, among others, [8, 11, 10]), and they are characterized by the fact that the underlying theory is modelled as a function whose output value is used to determine the class. *Tree-based* classification models are characterized by describing the underlying theory as a tree, and they range from *deterministic single-tree* models, such as ID3 or C4.5, to *random forest* models (see, e.g., [3, 18, 19]). Finally, *rule-based* classification models, which are of interest in this

*We thank the Italian INDAM GNCS project 'Formal methods for techniques for combined verification', and the Emilia-Romagna (Italy) regional project 'New Mathematical and Computer Science Methods for the Water and Food Resources Exploitation Optimization'.

Copyright © 2020 E. Lucena-Sánchez, E. Muñoz-Velasco, G. Sciavicco, I. E. Stan, and A. Vaccari
Space reserved for further notes in the final proceedings

work, describe an underlying theory by means of sets of rules, which are not necessarily dependent from each other, and, in a sense, imitate the human reasoning (see, among others, [12, 13, 16]). Rule-based classification generalizes tree-based classification, since a tree can be seen as a particular case set of rules, but not the other way around. Moreover, rule-based models are generally *interpretable*, that is, they are models that can be read (and explained) by a human, while function-based models (especially those based on neural networks) are not.

A common denominator to most classification models is the fact that they are *atemporal*. Classical classification problems consider static instances, in which the temporal component is absent, or neglected. In some cases, the temporal aspects of the information are abstracted (e.g., averaging the fever over all the observation period of the patients), so that static algorithms can be used. Nevertheless, in some applications, resorting to static models may not be adequate. Purely temporal instances such as in the above example can be seen as multivariate *time series*, that is, collections of points in time in which the interesting values are recorded: while there are several techniques devoted to single (univariate and multivariate) time series explanation and prediction, classification models for time series are still in their infancy. As shown in [20], time series can be abstracted into (interval-based) timelines in a rather general way, and timelines, in turn, can be described in a very convenient way by means of interval temporal logics, such as Halpern and Shoham's logic of Allen's relations (HS [9]). Static rule-based classification produces models that can be described by means of *propositional logic-like* rules; here, we propose an algorithm that is able to extract an *interval temporal logic-like* rule-based classifier, and in which finite interval temporal model checking [15] plays a central role. A tree-based classification model algorithm based on the same principle has been recently presented in [5], while a deterministic algorithm for HS *association* rule extraction from timelines has been proposed in [4].

2 The general context

There are several application domains in which learning a temporal classification model may be useful. Time information is commonly represented as a time series. The learning paradigm for the temporal case can be orthogonally partitioned in univariate versus multivariate learning, and reasoning about one instance versus a collection of instances. *Time series forecasting* [2] is the problem of predicting what will happen in the next unseen observation of an instance, given the finite temporal history of one variable. In the case of temporal multivariate learning with a single instance, one may try to describe the behaviour of a variable based on the values of other variables at the same time instant and/or previous ones. WEKA [21], among others, has packages for modelling such scenarios, where the time series are transformed in some convenient way so that any classical *propositional-like* model (e.g., multivariate regression) can be used to solve the problem. The cases when the problem is addressed to a collection of instances require a different treatment. When the analysis is centered on a single temporal variable, the typical solution proposed in the literature is to identify/mine *frequent patterns* across the instances to use as driving discriminant characteristics.

This work is part of a bigger project where the investigation is devoted to the more general problem of *multiple instances multivariate temporal classification*. Given a set of *labelled* (or *classified*) *instances* from a set of countable classes, the classification problem is to find a function/model that relates each instance to its class in such a way to minimize the misclassification error. The goal, in our case, is to treat time in an explicit way within the classification problem, by using the (well-studied) interval-based temporal logic formalism to *describe* an *interpretable theory* which takes full advantage of the temporal information, where the expressive power of the (interval) language plays a central role; in this sense, this problem generalizes the classical (static) classification problem (because of the temporal component) and the classical (temporal) forecasting problem (because it deals with multiple instances). A first step towards solving the multiple instances multivariate temporal classification problem has been done by devising a general algorithm to extract *timelines* (i.e., interval models) from time series, addressed in [20]. Such a representation allows an approach to abstract interval temporal learning, and highlights how the interval-based approach is, in general, more natural than a point-based one.

Inspired by the fact that interval temporal learning generalizes classical learning, it seems natural to

HS	**Allen's relations**	**Graphical representation**
$\langle A \rangle$	$[x,y]R_A[x',y'] \Leftrightarrow y = x'$	
$\langle L \rangle$	$[x,y]R_L[x',y'] \Leftrightarrow y < x'$	
$\langle B \rangle$	$[x,y]R_B[x',y'] \Leftrightarrow x = x', y' < y$	
$\langle E \rangle$	$[x,y]R_E[x',y'] \Leftrightarrow y = y', x < x'$	
$\langle D \rangle$	$[x,y]R_D[x',y'] \Leftrightarrow x < x', y' < y$	
$\langle O \rangle$	$[x,y]R_O[x',y'] \Leftrightarrow x < x' < y < y'$	

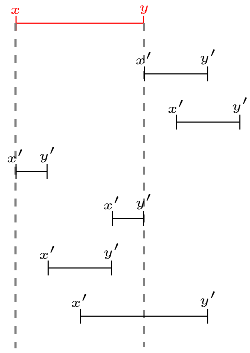

Figure 1: Allen's interval relations and HS modalities.

approach this problem by generalizing classical (propositional-like) algorithms. A preliminary result in this sense has been presented in [5], in the form of an interval-based generalization of the well-known decision tree induction algorithm ID3. In this work, we consider the problem of extracting a temporal classifier composed of a set of implicative rules following the same principle: we start from a temporal data set of labelled timelines and we extract a temporal classifier (in which the left-hand side of rules is written in interval temporal logic) by generalizing a classical extraction algorithm.

3 Background

Interval temporal logic. Let $\mathbb{D} = \langle D, < \rangle$ be a strict linear order. An *interval* over $\mathbb{D}$ is an ordered pair $[x, y]$, where $x, y \in D$ and $x < y$. We denote the set of all intervals over $\mathbb{D}$ with $\mathbb{I}(\mathbb{D})$. There are 13 different Allen's relations between two intervals in a linear order [1]: the six relations R_A (adjacent to), R_L (later than), R_B (begins), R_E (ends), R_D (during), and R_O (overlaps), depicted in Fig. 1, and their inverses, for each $X \in \mathcal{A}$, where $\mathcal{A} = \{A, L, B, E, D, O\}$. Halpern and Shoham's modal logic of temporal intervals (HS) is defined from a set of propositional letters AP, and by associating a universal modality $[X]$ and an existential one $\langle X \rangle$ to each Allen's relation R_X. Formulas of HS are obtained by:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle\varphi \mid \langle \overline{X} \rangle\varphi \mid [=]\varphi,$$

where $p \in \mathrm{AP}$ and $X \in \mathcal{A}$. The other Boolean connectives and the logical constants, as well as the universal modalities $[X]$, can be defined in the standard way. Observe that the operator $[=]$ is redundant and not included in the original language; it becomes important in the phase of inducting a formula, as we shall see. The semantics of HS formulas is given in terms of *timelines* $T = \langle \mathbb{I}(\mathbb{D}), V \rangle$, where $V : \mathrm{AP} \to 2^{\mathbb{I}(\mathbb{D})}$ is a *valuation function* which assigns to each atomic proposition $p \in \mathrm{AP}$ the set of intervals $V(p)$ on which p holds. The *truth* of a formula φ on a given interval $[x, y]$ in an interval model T is defined by structural induction on formulas as follows:

$$\begin{array}{lll}
T,[x,y] \Vdash p & \text{if} & [x,y] \in V(p), \text{ for } p \in \mathrm{AP}; \\
T,[x,y] \Vdash \neg\psi & \text{if} & T,[x,y] \nVdash \psi; \\
T,[x,y] \Vdash \psi \vee \xi & \text{if} & T,[x,y] \Vdash \psi \text{ or } T,[x,y] \Vdash \xi; \\
T,[x,y] \Vdash \langle X \rangle\psi & \text{if} & \text{there is } [w,z] \text{ s.t } [x,y]R_X[w,z] \text{ and } T,[w,z] \Vdash \psi, \\
T,[x,y] \Vdash \langle \overline{X} \rangle\psi & \text{if} & \text{there is } [w,z] \text{ s.t } [x,y]R_{\overline{X}}[w,z] \text{ and } T,[w,z] \Vdash \psi, \\
T,[x,y] \Vdash [=]\psi & \text{if} & T,[x,y] \Vdash \psi.
\end{array}$$

In [20], a general-purpose algorithm has been presented that is able to transform multivariate time series into timelines (i.e., a *temporal data set*), which can be used to learn a classification model. Temporal data sets have been first introduced in [15], where a polynomial time finite model checking algorithm for formulas of HS has been proposed.

Learning a rule-based supervised classifier. Given a data set $\mathcal{S}$ of m *instances* $I_1, \ldots, I_m$, each represented by n distinct *attributes*, and each belonging to a predetermined *class*, the *supervised classification model extraction* (or *learning*) is the task of inducing a model that explains how the values of the attributes determine the value of the class [14, 17]. A *rule-based* classification model is characterized by the set of its rules (e.g., if the patient is over 40, then he/she presents a recurrence). Even if static instances present numeric attributes, classical classification models are considered *propositional*. A typical propositional rule has the form:

$$\rho : p_1 \wedge p_2 \wedge \ldots \rightarrow c, \tag{1}$$

where $A(\rho) = p_1 \wedge p_2 \wedge \ldots$ is its *antecedent*, and $C(\rho) = c$ is its *consequent*, and a rule-based model has the form:

$$\Gamma = \begin{cases} \rho_1 : p_1^1 \wedge p_2^1 \wedge \ldots p_{s_1}^1 \rightarrow c_1 \\ \rho_2 : p_1^2 \wedge p_2^2 \wedge \ldots p_{s_2}^2 \rightarrow c_2 \\ \ldots \\ \rho_k : p_1^k \wedge p_2^k \wedge \ldots p_{s_k}^k \rightarrow c_k \end{cases} \tag{2}$$

Each proposition p_l^h above represents a condition (such as the patient being over 40); even if the symbols c_l do not belong to the object language, the rules can be considered as written in propositional logic (in particular, $A(\rho)$, that is, the antecedent of ρ, is a pure propositional formula). Rules have the form of logical implications, but they are not interpreted as such: the notion of a rule being *satisfied* on a data set $\mathcal{S}$ (composed by several instances) is statistical, rather than logical. In particular, given an instance $I \in S$, we say that a rule ρ *holds* on I if $A(\rho)$ is satisfied by the values of I, and we say that I is *classified as c by* Γ (denoted $\Gamma(I) = c$) if, given the rules of Γ that hold on I, and given the *firing policy* of Γ, the class c is assigned to I. There are standard ways to evaluate the *performances* of a supervised classifier.

4 An Optimization Model for Inducing Temporal Logic Rule-Based Classifiers

Let us consider a temporal data set $\mathcal{T}$ of timelines $T_1, T_2, \ldots, T_m$. Each timeline is the abstraction of a multivariate time series with n distinct variables, and it is (binary) classified. We define a generic *temporal classification rule* of HS as an object ρ of the type:

$$\rho : p_1 \wedge Op(p_2 \wedge Op(\ldots) \ldots) \rightarrow c, \tag{3}$$

where each p_i is a propositional letter or the constant $\top$, and each Op is either $\langle X \rangle, [X], \langle \overline{X} \rangle$, or $[\overline{X}]$, where $X \in \mathcal{A}$. While there are several possible alternative forms for temporal classification rules, an object of type of (3) is immediately interpretable, allows rules to take the form of (temporal) patterns, and generalizes (1). For us, a *temporal classifier* Γ is an object of the type:

$$\Gamma = \begin{cases} \rho_1 : p_1^1 \wedge Op(p_2^1 \wedge Op(\ldots \wedge p_{s_1}^1) \ldots) \rightarrow c_1 \\ \rho_2 : p_1^2 \wedge Op(p_2^2 \wedge Op(\ldots \wedge p_{s_2}^2) \ldots) \rightarrow c_2 \\ \ldots \\ \rho_k : p_1^k \wedge Op(p_2^k \wedge Op(\ldots \wedge p_{s_k}^k) \ldots) \rightarrow c_k. \end{cases} \tag{4}$$

Now let $\bar{u}$ be a vector of decision variables (a *candidate* solution) that encodes a classifier, and let $\Gamma_{\bar{u}}$ (resp., $\bar{u}_\Gamma$) be the classifier that corresponds to $\bar{u}$ (resp., the encoding that corresponds to Γ). The classifier $\Gamma_{\bar{u}}$ can be evaluated on the temporal data set $\mathcal{T}$ to obtain a measure $Per(\Gamma_{\bar{u}})$ of its performance. Moreover, let $Dim(\Gamma)$ be a function measuring the total number of symbols used in Γ, $md(\Gamma)$ a function measuring the maximum modal depth of any antecedent $A(\rho)$ for any $\rho \in \Gamma$, and $|\Gamma|$ a function that measures the number of rules in Γ. Then, the problem of inducing a temporal rule-based classifier for a given temporal data set can be seen as the following *multi-objective optimization problem* (see, e.g. [6]):

Algorithm 1: Rule-based classifier extraction algorithm.

input : $\Gamma, \mathcal{T}, \max_\rho$, and $\max_{md}$.

$\Pi \leftarrow InitialPopulation(\max_\rho, \max_{md}, \mathrm{AP})$ `// random population of candidate solutions`
for $i = 1$ **to** $maxGen$ **do**
  foreach $\Gamma \in \Pi$ **do**
    $Per(\Gamma) \leftarrow ComputePerformance(\Gamma, \mathcal{T})$ `// using finite model checking [15]`
    $Dim(\Gamma) \leftarrow ComputeDimension(\Gamma)$
  end
  $\Pi' \leftarrow Selection(\Pi, Per(\Gamma), Dim(\Gamma))$ `// (Pareto) non-dominated solutions`
  $\Pi \leftarrow NewGeneration(\Pi')$ `// new population via evolutionary operators`
end
return Π

$$\begin{cases} \max \;\; Per(\Gamma_{\bar{u}}) \\ \min \;\; Dim(\Gamma_{\bar{u}}) \\ \textit{subject to} \\ 2 \leq |\Gamma_{\bar{u}}| \leq \max_\rho \\ md(\Gamma_{\bar{u}}) \leq \max_{md}, \end{cases} \tag{5}$$

where $\max_\rho$ and $\max_{md}$ are, respectively, the maximum number of rules in classifier and the maximum modal depth for each rule. *Multi-objective evolutionary algorithms* (see, e.g., [7]) are known to be particularly suitable to perform multi-objective optimization, as they search for multiple optimal solutions in parallel. Algorithm 1 is the adaptation of the general schema of a evolutionary algorithm to our case, in which we abstract over the difference between a solution Γ and its internal representation $\bar{u}_\Gamma$. After *maxGen* generations, the procedure stops. The solution of multi-objective optimization is a population of candidates that tends towards the the so-called *Pareto* front. This means that, in general, we shall have many possible classifiers. Each classifier is *optimal* within the part of the search space that has been explored, that is, during the execution, all classifiers that have been produced with worse performance and same dimension, as well as all classifiers with greater dimension and same performance have been discarded. From the returned population, a *decision-making* strategy must be applied to choose one classifier, and that may depend from the intended application.

5 Conclusions

Supervised classification is one of the main computational tasks of modern Artificial Intelligence. In this paper we defined the problem of supervised temporal classification, and then we proposed a solution based on interval temporal logic, essentially defining the task of temporal classification model induction as an optimization problem in which finite interval temporal model checking plays a central role.

References

[1] J. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[2] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time Series Analysis: Forecasting and Control.* Wiley, 5 edition, 2015.

[3] L. Breiman. Random forests. *Machine Learning*, 1(45):5–32, 2001.

[4] D. Bresolin, E. Cominato, S. Gnani, E. Muñoz-Velasco, and G. Sciavicco. Extracting interval temporal logic rules: A first approach. In *Proc. of the 24th International Symposium on Temporal Representation*

and Reasoning (TIME), number 120 in Leibniz International Proceedings in Informatics, pages 1–15, 2018.

[5] A. Brunello, G. Sciavicco, and I. Stan. Interval temporal logic decision tree learning. In *Proc. of the 16th European Conference on Logics in Artificial Intelligence (JELIA 2019)*, number 11468 in Lecture Notes in Computer Science, pages 778–793, 2019.

[6] Y. Collette and P. Siarry. *Multiobjective Optimization: Principles and Case Studies.* Springer Berlin Heidelberg, 2004.

[7] K. Deb. *Multi-objective optimization using evolutionary algorithms.* Wiley, London, UK, 2001.

[8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* MIT Press, 2016.

[9] J. Halpern and Y. Shoham. A propositional modal logic of time intervals. *Journal of the ACM*, 38(4):935–962, 1991.

[10] S. Haykin. *Neural networks: A comprehensive foundation.* Prentice Hall, 1999.

[11] D. Hosmer and S. Lemeshow. *Applied Logistic Regression (2nd ed.).* Wiley, 2000.

[12] J. Ji, N. Zhang, C. Liu, and N. Zhong. An ant colony optimization algorithm for learning classification rules. In *Proc. of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 1034–1037, 2006.

[13] Y. Liu, Q. Zheng, Z. Shi, and J. Chen. Rule discovery with particle swarm optimization. In *Proc. of the 2004 Conference on Advanced Workshop on Content Computing (AWCC)*, volume 3309 of *Lecture Notes in Computer Science*, pages 291–296, 2004.

[14] T. Mitchell. *Machine Learning.* McGraw-Hill, New York, NY, USA, 1997.

[15] D. D. Monica, D. de Frutos-Escrig, A. Montanari, A. Murano, and G. Sciavicco. Evaluation of temporal datasets via interval temporal logic model checking. In *Proc. of the 24th International Symposium on Temporal Representation and Reasoning (TIME)*, pages 1–18, 2017.

[16] M. Muntean, C. Rotar, I. Ileană, and H. Vălean. Learning classification rules with genetic algorithm. In *Proc. of the 8th International Conference on Communications*, pages 213–216, 2010.

[17] K. Murphy. *Machine learning: a probabilistic perspective.* MIT press, 2012.

[18] J. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[19] J. Quinlan. Simplifying decision trees. *International Journal of Human-Computer Studies*, 51(2):497–510, 1999.

[20] G. Sciavicco, I. Stan, and A. Vaccari. Towards a general method for logical rule extraction from time series. In *Proc. of the 8th International Work-conference on the Interplay between Natural and Artificial Computation (IWINAC)*, number 11487 in Lecture Notes in Computer Science, pages 3–12, 2019.

[21] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques.* Wiley, 3 edition, 2015.