Project Description
CPM: Cyber-Physical Data Mining for Predictive Maintenance

Oded Maler

June 23, 2017

1 Scientific Context

The world around us is in a constant flux with "things" changing dynamically. Houses are air-conditioned, power is generated, distributed and consumed, cars drive on roads and highways, plants manufacture materials and objects, commercial transactions are made and recorded in information systems. Airplanes fly, continuously changing location and velocity while their controllers deal with various state variables in the engine and wings. Those processes can be viewed as generating *temporal behaviors* (*waveforms*, *signals*, *time series*, *sequences*) where continuous and discrete variables change their values and various types of events occur along the time axis.

The systems that generate these behaviors are evaluated to some extent as good or bad, efficient or wasteful, excellent or catastrophic. We use *monitoring* to denote the act of observing and evaluating such temporal behaviors. Behaviors can be very long, spanning over a large stretches of (possibly continuous) time, densely populated with observations. They can also be very wide, recording many variables and event types. As such they carry too much information by themselves to be easily and directly evaluated. What should be distilled out of these behaviors should somehow be expressed and specified. The mathematical objects that do this job are functions that map complex and information-rich behaviors into low dimensional vectors of bits and/or numbers that indicate satisfaction of logical and numerical constraints or the values of various performance indices.

In recent years, several formalisms such as *signal temporal logic* (STL) [10, 11, 13] and *timed regular expressions* (TRE) [3, 14] have been defined to specify properties and measurements of such behaviors. They can be viewed as yet another way to evaluate continuous-time signals that complements traditional measures used in statistics, control and signal processing. These formalisms have been applied successfully to check behaviors produced by simulators and automatically detect property violations in a variety of domains including embedded control systems, robots, analog circuits, music and biochemical reactions. Most of these applications were characterized by two features: 1) They were applied in the development and verification stage and hence dealt with virtual *simulated* behaviors; 2) They solved the *direct* problem: does a behavior satisfy a *given* specification. In this project we will relax these two assumptions (see also the attached discussion in [9]).

First we will be interested in monitoring *real* systems during their execution. In this context, the role of temporal specification changes: it should express alarming patterns that require some automatic or human reaction *before* it is too late. There is a plethora of application domains for this *predictive maintenance* activity: to detect health degradation of humans and machines, suspicious internet activity, congestions in physical and information highways and more. The role of an expressive

and semantically correct pattern description language cannot be under-estimated here. Secondly, we will attack the *inverse* problem: come up with a formal description (STL formula or TRE) compatible with observed signals. This is a problem of machine learning or data mining, that we intend to attack in a synergetic way, using concepts from formal verification and monitoring and adapting them the the noisy context of learning from real data. Such a technique has numerous applications such as 1) building an abstract high-level model from execution traces of a system for the purpose of more efficient simulation and compositional verification; 2) the automatic derivation of specification [7] based on positive and negative examples of executions provided by an expert; and 3) the unsupervised clustering of observed behaviors of new technological devices [5]. In the context of predictive maintenance the big challenge is to identify temporal conditions that precede undesired and catastrophic events.

The project will thus export concepts developed in formal verification to the domain of machine learning, using temporal logic and regular expressions as a new class of *feature extractors* that specialize in the sequential aspects of behaviors.

2 Scientific Program

2.1 Algorithms for Parametric Identification

The starting point of the project will be based on parametric extensions of STL and signal regular expressions (SRE which are TRE with predicates over real-values variables). We will mostly consider the case where all parameters have a fixed polarity, that is, if $p < p'$ it is always easier (or always harder) to satisfy φ_p than $\varphi_{p'}$. Under these assumption, the set of parameters that render a formula satisfied by a given trace or set of traces can be made upward-closed (in the partial-order on the parameter space) and it is sufficient to find the set of minimal (tightest) parameters that lead to satisfaction. In [4] a preliminary effort to approximate this set (which has the structure of a Pareto front) has been made using a very simple search procedure. In this project we will first develop a more powerful algorithm for Pareto front approximation to be able to handle parameter spaces of up to 10 dimensions. Such a procedure, which is a multi-dimensional generalization of binary search, is interesting by itself.

Once this procedure is in place we will apply it to parametric identification based on some commonly used templates of formulas and expressions, for example, the property of time-bounded stabilization $\varphi = e \rightarrow F_{[0,a]} x < c$ which states in STL the fact that within a time after event e, the value of variable x drops below c. Finding the minimal pairs $p = (a, c)$ that render φ_p satisfied by a set of traces gives a valuable information. We will explore additional formula templates, admitting progressively more parameters (for example the time duration such that x remains below c), and identify their parameters to come up with abstract descriptions of the system that generates these behaviors.

We will explore the limits of the procedure trying to complement it by alternative techniques, for example finding the tightest parameters using quantifier elimination (when the signals are linearly interpolated) or find a more continuous formulation of parameter dependencies that will allow us to apply gradient-based rather then search-based methods [2].

2.2 Extensions

2.2.1 Non Parametric Identification

While parametric identification based on a single template is efficient, sometimes we do not know a priori the form of the formula which then becomes an additional part of the search space. This

immediately raises the risk of combinatorial explosion and a special care should be taken to restrict the class of formulae considered, for example to be a size-bounded Boolean combination of a fixed set of templates. The choice of templates and their combination will be strongly influenced by the case-studies. Another way to achieve non-parametric identification is to use automaton learning techniques. We have recently applied Angluin's automaton learning algorithm to numerical alphabets [12] but a lot needs to be done to adapt the premises of this algorithm (helpful teacher, active learning, noise freedom) to real life situations.

2.2.2 Treating Noise and Outliers

In the previously described approach, p is considered a valid parameter only if φ_p is satisfied by *all* the traces and for all their duration. Given that real-life data are noisy, we can relax this requirement and associate with each parameter value a real number indicating the fraction of traces that satisfy φ_p. Under the assumption of fixed polarity parameters, this function is monotone and learning an approximation of it is the quantitative analogue of approximating a Parteto front. We will develop a specialized approximate learning algorithm for this class of functions and use its form as a basis for signal classification.

Another related direction is to move from satisfaction to matching. The former tells us whether the whole behavior satisfies a property, while the latter identifies the segments of the behavior where the patterns occur or are violated. Thus, for example, a property that deals with response time (distance between events) can be satisfied in certain periods of time and violated during rush hour. We will extend the pattern matching algorithm of [14] for TRE to deal also with STL and to indicate the quantitative robustness of the satisfaction [6].

2.3 Learning Alarming Conditions

This is the most challenging part of the project and its progress will depend on the selection of the case-studies among several domains described in the next sub-section, and on the feed-back obtained from them. We will consider execution logs or simulation traces in which some highly undesirable event e occurs, like a patient having a heart attack or a processor in the cloud dropping dead. We would like to come up with a condition of the form $\varphi \rightarrow F_{[a,b]}e$ where φ is some formula whose satisfaction at time t implies the occurrence of the bad event sometime in $[t+a, t+b]$. When such a formula is identified, it can be the object of monitoring and alarm generation. Let us mention that recently we applied similar ideas to the detection of "activity triggers", events in a complex digital circuit that activate and de-activate certain components [8].

Similar ambitions have been raised, of course, in many contexts of machine learning and have been attacked using traditional techniques such as auto-regression (ARMA) models that relate signal values at different time instances, or recurrent neural networks. One of the outcomes of the project will be a comparison of the newly proposed approach with those existing techniques, finding the respective merits of each. The approach advocated in this project brings tools, inspired by verification technology, that specialize in the *temporal*, *sequential* aspects of the phenomenon and which handle dense time naturally. Our experience in automaton learning over numerical alphabets [12] may suggest that the sequential aspects is best handled explicitly while the width data aspects can be handled by various static learning techniques, deep or not.

2.4 Online Monitoring

Offline monitoring of simulation traces can be done in any future-past order while online monitoring of a real system must be aligned with the progress of time. Part of the project will involve the improvement of the online algorithms for STL [11] and TRE [15] and their adaptation to the specific requirements of timely alarm generation. The topic is also related to theoretical issues concerning the relation between to future and past fragments of temporal logic.

2.5 Applications

In the initial phase of the project we will explore several sources of data to which our techniques will be later applied. Among the data sets that we consider we mention:

- Automotive: our ongoing collaboration with Toyota can lead to two classes of data sets. The first are simulated data from power train engine models. The second are logs of test drives for new fuel cell (hydrogen technology). Such data have been used recently to do STL-based clustering [5];
- Medical: data from cardiac applications where regular expressions have been recently used as a pattern description language [1];
- Information technology: execution logs of computers in data centers to detect conditions that precede to processor faults.
- Other possible data sets provided by industrial collaborators

3 Project Organization

In the initial phase, a series of research seminars common to the VERIMAG and LIG teams will be conducted to achieve mutual transfer of knowledge. VERIMAG will present the underlying concepts of formal specification and monitoring while LIG will present the relevant parts of the state-of-the-art in learning time series and temporal behaviors.

Then we will progress in two parallel directions. First we will develop and improve the algorithmics of parametric identification so as to apply it to real-life application. Secondly we will investigate the potential case-studies, identify their interesting properties and patterns and characterize practical aspects such as size and noisiness. A subset of those will be selected for the application of the developed techniques and will accompany the project.

Then we intend to investigate the form of the alarming conditions appropriate for these application domains and tune the identification algorithms to handle them, using new extensions whose specifics will be unfolded during the investigations. Since this is a research project, a more detailed plan would be a wishful thinking. The post-doc will share his/her time between VERIMAG and LIG and will also interact with the external collaborators.

References

[1] Houssam Abbas, Alena Rodionova, Ezio Bartocci, Scott A Smolka, and Radu Grosu. Regular expressions for irregular rhythms. *arXiv preprint arXiv:1612.07770*, 2016.

[2] Houssam Abbas, Andrew Winn, Georgios Fainekos, and A Agung Julius. Functional gradient descent method for metric temporal logic specifications. In *2014 American Control Conference*, pages 2312–2317. IEEE, 2014.

[3] Eugene Asarin, Paul Caspi, and Oded Maler. Timed regular expressions. *Journal of the ACM*, 49(2):172–206, 2002.

[4] Eugene Asarin, Alexandre Donzé, Oded Maler, and Dejan Nickovic. Parametric identification of temporal properties. In *Runtime Verification*, pages 147–160. Springer, 2011.

[5] Jyotirmoy V Deshmukh, Xiaoqing Jin, Sanjit Seshia, et al. Learning auditable features from signals using unsupervised temporal projection. In *HSCC*, 2017.

[6] Alexandre Donzé and Oded Maler. Robust satisfaction of temporal logic over real-valued signals. In *FORMATS*, pages 92–106, 2010.

[7] Xiaoqing Jin, Alexandre Donzé, Jyotirmoy V. Deshmukh, and Sanjit A. Seshia. Mining Requirements from Closed-loop Control Models. In *HSCC*, 2013.

[8] Jan Lanik, Legriel Julien, E Piriou, E Viaud, Fahim Rahim, Oded Maler, and S. Rahim. Reducing power with activity trigger analysis. In *NEMOCODE*. 2015.

[9] Oded Maler. Some thoughts on runtime verification. In *International Conference on Runtime Verification*, pages 3–14, 2016.

[10] Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *FORMATS/FTRTFT*, pages 152–166, 2004.

[11] Oded Maler, Dejan Nickovic, and Amir Pnueli. Checking temporal properties of discrete, timed and continuous behaviors. In *Pillars of Computer Science*, pages 475–505, 2008.

[12] Irini-Eleftheria Mens and Oded Maler. Learning regular languages over large ordered alphabets. *Logical Methods in Computer Science (LMCS)*, 11(3), 2015.

[13] Dejan Nickovic. *Checking timed and hybrid properties: Theory and applications*. PhD thesis, Université Joseph Fourier, Grenoble, France, 2008.

[14] Dogan Ulus, Thomas Ferrère, Eugene Asarin, and Oded Maler. Timed pattern matching. In *FORMATS*, pages 222–236, 2014.

[15] Dogan Ulus, Thomas Ferrère, Eugene Asarin, and Oded Maler. Online timed pattern matching using derivatives. In *TACAS*, pages 736–751, 2016.