The 29^{th} ILP conference. 3–5 Sep 2019, Plovdiv, Bulgaria.

Learning to Detect Complex Events with Expert Advice

Nikos Katzouris

Institute of Informatics, NCSR “Demokritos”, Athens, Greece
nkatz@iit.demokritos.gr

Abstract. Systems for symbolic event recognition detect occurrences of events in streaming input using a set of event patterns in the form of temporal logical rules. Algorithms for online learning/revising such patterns should be capable of updating the current event pattern set without compromising the quality of the provided service, i.e. the system’s online predictive performance. Towards this, we present an approach based on Prediction with Expert Advice. The experts in our approach are logical rules representing event patterns, which are learnt online via a single-pass strategy. To handle the dynamic nature of the task, an Event Calculus-inspired prediction/event detection scheme allows to incorporate commonsense principles into the learning process. We present a preliminary empirical assessment with promising results.

Keywords: Online Relational Learning · Complex Event Recognition.

1 Introduction

Complex event recognition (CER) systems [3] process streams of *simple events*, (e.g. sensor data) and recognize *complex events* (CEs), i.e. events that satisfy some pattern. Systems for symbolic CER [1] typically use logical rules to represent CE patterns. Algorithms for online learning/revising such patterns are highly desirable, since useful patterns may not always be available beforehand, while existing ones often need to be updated to reflect change in the data. In this respect, logic-based CER frameworks have the advantage of allowing direct connections to machine learning via ILP.

Algorithms that learn from event streams should operate in a single pass over the input, while continuously making accurate predictions, in our case, detecting CEs. Existing online learners for CER applications [5, 6], however, focus more on the single-pass nature of the task and less on online performance, which is not optimized and is therefore sub-optimal. We address this issue by using Prediction with Expert Advice (PEA) [2], an online learning framework for making accurate predictions via combining the “advice” of an ensemble of “experts” (independent predictors), while guaranteeing predictive performance comparable to that of the best expert (or combination thereof). The experts in our case are rules representing CE patterns, generated online using a single-pass rule learning strategy adopted from the OLED algorithm [5], which uses Hoeffding bounds to assess the quality of the rules from small subsets of the input.

Temporal domains, such as CER, are dynamic in nature and often characterized by commosense phenomena. Building temporal reasoning engines using action formalisms, such as the Event Calculus [7], is a way to incorporate commonsense principles into a CER system [1]. Inspired from such approaches, our proposed algorithm is equipped with a prediction/event detection scheme that takes into account the persistence of detected CEs in time.

In this extended abstract, we present a high-level description of our approach, along with a preliminary experimental evaluation, which indicates that the new algorithm yields significant improvements over state-of-the-art algorithms.

2 Background

The Event Calculus (EC) [7] is a temporal logic for reasoning about events and their effects. Its ontology comprises fluents, i.e. properties which have certain values in time; and events, i.e. occurrences in time that may affect fluents and alter their value. The axioms of the EC incorporate the *common sense law of inertia*, according to which fluents persist over time, unless they are affected by an event. In this work, fluents correspond to CEs and CE patterns to initiation and termination rules, which specify the dynamics of the domain. The learning setting assumes a relational stream of observations, labelled with CE instances, from which we wish to learn, in an online fashion, a set of initiation and termination conditions for the CEs. We refer to [6] for a detailed account of the EC dialect that we employ and a concrete example.

In **Prediction with Expert Advice (PEA)** [2] a "master" attempts to predict the value of a variable, by combining the predictions of an ensemble of "experts". Each expert carries a weight (a non-negative real number). Learning proceeds in rounds. At each round the master casts its prediction and then the true outcome is revealed. The master and each expert suffer a loss, and the weights of the experts are updated in a multiplicative manner, in proportion to their loss. PEA algorithms come with guarantees on the master's *regret* i.e., the difference between the the master's total loss over a sequence of rounds and the loss of the best expert in hindsight. The experts in our approach are initiation and termination rules in the EC, and the prediction task refers to inferring the truth values of CE instances over time. Using rule-based experts calls for a generalization of the PEA framework, called *sleeping experts* [4]. In this setting a rule expert makes a prediction only on rounds where it "fires" (awake expert), otherwise it abstains (asleep expert).

3 The Approach

At time t the algorithm maintains an ensemble of rules, receives a set of observations and predicts the truth value of a set of query atoms (CE instances).

Making Predictions & Updating Weights: The way that the experts predict mimics the functionality of initiation and termination axioms in the EC. Awake initiation experts consistently predict 1 (true) for the query atom $q =? \mathsf{holdsAt}(\alpha, t+1)$ (where α is a CE instance), since at the current time t such rules initiate α, causing it to hold at query time $t+1$. Awake termination experts predict 0 (false), since such rules terminate α causing it to not hold at $t+1$.

The master combines the experts' predictions via a weighted average scheme, returning confidence-rated predictions in $[0, 1]$. An additional "inertia expert" uses these confidence-rated master's predictions to replicate the functionality of inertia in the

Event Calculus. Once the master predicts that a CE instance α holds at some time point, the inertia expert "remembers" α along with the associated confidence of the prediction. In subsequent rounds the inertia expert makes weighted predictions on the truth value of α (predicting 1 with the associated weight). As with all other experts, the inertia expert's weight w.r.t. a CE instance α is demoted with incorrect predictions, and the expert "remembers" α for as long as its weight exceeds a given threshold, otherwise it "forgets" it. From that point on, the inertia expert is considered "asleep" (does not make predictions) w.r.t. α, until such time when α is re-initiated at some future round.

We omit technical details due to space limitations and simply note that the master's weighted average prediction scheme, as well as the multiplicative weight update strategy follow the classical sleeping experts algorithm [4]. The weights update strategy is such that it rewards a correct expert, especially when the master had a high probability of predicting incorrectly, and penalizes an incorrect expert, especially when the master had a high probability of predicting correctly.

Structure Learning: The rule/expert structure update strategy consists of three operations: (a) ensemble expansion, which augments the current ensemble with new rules in response to erroneous predictions of the master; (b) rule expansion, which progressively constrains over-general rules by adding extra conditions to their bodies; and (c) ensemble pruning, which removes under-performing rules with a very low weight.

The ensemble expansion operation is triggered by the absence of correct experts during a master's prediction. In such cases, a new set of experts is generated, by encoding the incorrectly predicted example into a bottom clause and searching for rules that θ-subsume it. The rule expansion operation corresponds to a hill-climbing search into the subsumption lattice, where at each step a rule is specialized (expanded) by the addition of one literal to its body, in a top-down, FOIL-like process. The hill-climbing search is single-pass by using Hoeffding bounds, similarly to the OLED algorithm [5].

4 Experiment

We present a preliminary evaluation on the CAVIAR[1] dataset for activity recognition, see [6] for details on the dataset and the domain. There are two target CEs in CAVIAR, related to two persons *moving* together and *meeting* each other. We compare OLED-EXP (the new PEA-based algorithm) to the following approaches: (i) HC, a hand-crafted set of rules; (ii) HC-EXP, the hand-crafted rule set with weights learnt by the sleeping experts algorithm; (iii) OLED [5]; (iv) OLED-MLN [6], a Markov Logic Networks (MLN)-based version of OLED that uses the AdaGrad for weight learning. We used prequential evaluation, the standard online setting where an incoming example is first used to test the current model and then to update it. The results are presented in Figure 1, in terms of the number of accumulated mistakes over time – false positives and false negatives.

Thanks to its ability to quickly identify the best expert from those currently available, while progressively improving the experts' performance by modifying their structure, OLED-EXP achieves significant improvements, as compared to the other approaches. OLED's performance is the worst among all learners, comparable to that of the hand-crafted rule set. This was expected, as OLED simply learns structure in a single-pass,

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

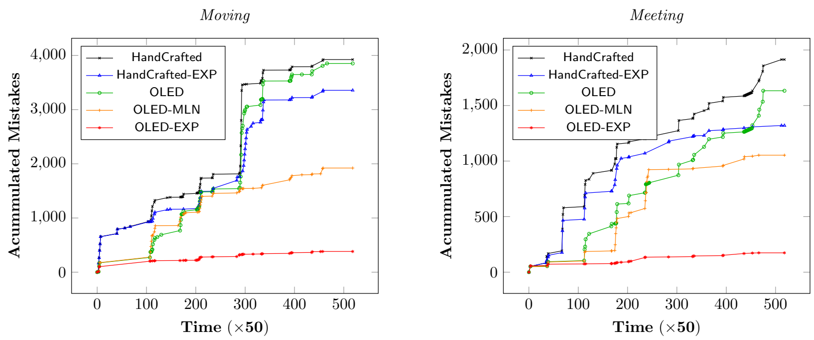

Fig. 1: Online prequential evaluation on CAVIAR.

without attempting to optimize the online performance of the evolving patterns. This is not the case for OLED-MLN, which supports weight learning. It achieves the second-best performance, however, with a significant distance from OLED-EXP. We attribute this result to the additive nature of OLED-MLN's weight updates. It is known that additive algorithms may exhibit worse online performance, as compared to multiplicative ones, especially in cases where a relatively small number of features is highly informative. This is typically the case in relational learning domains, where the number of useful rules is a small fraction of the total number of the generated rules.

5 Conclusions and Next Steps

We presented a novel algorithm for learning CE definitions in an online fashion, along with a preliminary experimental evaluation with promising results. Next steps involve a more thorough empirical assessment, including concept drift experiments.

References

1. Alexander Artikis, Marek Sergot, and Georgios Paliouras. An event calculus for event recognition. *Knowledge and Data Engineering, IEEE Transactions on*, 27(4):895–908, 2015.
2. Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.
3. Gianpaolo Cugola and Alessandro Margara. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*, 44(3):15, 2012.
4. Yoav Freund, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. Using and combining predictors that specialize. In *ACM Symposium on the Theory of Computing*, pages 334–343, 1997.
5. Nikos Katzouris, Alexander Artikis, and Georgios Paliouras. Online learning of event definitions. *TPLP*, 16(5-6):817–833, 2016.
6. Nikos Katzouris, Evangelos Michelioudakis, Alexander Artikis, and Georgios Paliouras. Online learning of weighted relational rules for complex event recognition. In *ECML-PKDD 2018*, pages 396–413, 2018.
7. Robert Kowalski and Marek Sergot. A logic-based calculus of events. *New Generation Computing*, 1986.