

Learning Complex Event Descriptions by Abstraction

Ugo Galassi¹, Attilio Giordana¹, Lorenza Saitta¹ and Marco Botta²

¹Università Amedeo Avogadro, Dipartimento di Informatica
Via Bellini 25/G, 15100 Alessandria, Italy

²Università di Torino, Dipartimento di Informatica
C.so Svizzera 185, 10149 Torino, Italy

{galassi, attilio, saitta}@mfn.unipmn.it, botta@di.unito.it

Abstract

The presence of long gaps dramatically increases the difficulty of detecting and characterizing complex events hidden in long sequences. In order to cope with this problem, a learning algorithm based on an abstraction mechanism is proposed: it can infer a Hierarchical Hidden Markov Model, from a learning set of sequences. The induction algorithm proceeds bottom-up, progressively coarsening the sequence granularity, and letting correlations between subsequences, separated by long gaps, naturally emerge. As a case study, the method is evaluated on an application of user profiling. The results show that the proposed algorithm is suitable for developing real applications in network security and monitoring.

1 Introduction

This paper addresses the task of discovering complex events (CEs) occurring sparsely in long sequences. It is assumed that a CE is a partially ordered set of short chains (episodes) of atomic events (AEs), interleaved with gaps, where irrelevant facts may occur. Moreover, the presence of noise can make episodes hard to recognize. Episodes are represented as strings of symbols, being a symbol the label assigned to an atomic event. In a recent paper [Botta *et al.*, 2004], a method for automatically inferring a Hierarchical Hidden Markov Model (HHMM) [Fine *et al.*, 1998] from a database of sequences has been proposed. Here, an improved version of the algorithm is applied to a non trivial application of user profiling in computer security.

2 User Profiling

User profiling is widely used to detect intrusions in computer networks or in telephony networks. The possibility of automatically building a profile for *users* or for *network services* reflecting their temporal behavior would offer a significant help to the deployment of adaptive Intrusion Detection Systems (IDSs) [Lee *et al.*, 2002].

The experiments described in the following investigate the possibility of automatically constructing a user profile from

the logs of its activity. The selected task consists in learning to identify a user from his/her typing style on a keyboard. The basic assumption is that every user has a different way of typing, which becomes particularly evident when he/she types specific words, or sequences of characters. The goal is not challenging the results previously obtained [Bleha *et al.*, 1990; Brown and Rogers, 1993], but investigating the possibility of synthesizing automatically profiles from the activity logs. The specific application has been selected because the data are easy to acquire. In other words, if the methodology described so far succeeds in building up a HHMM for this kind of user profiling, it is likely that it will succeed in other cases as well. Two experiments, described in the following subsections, have been performed.

3 Key Phrase Typing Model

In the first experiment, the goal was to construct a model for a user typing a *key phrase*, discriminant enough to recognize the user among others. A selected sentence of 22 syllables has been typed many times on the same keyboard, while a transparent program recorded the duration of each stroke, and the delay between two consecutive strokes. Then, every repetition of the sentence generated a temporal sequence. Four volunteers provided 140 sequences each, and, for every one of them, a model has been built up using 100 traces (for each user) as learning set. The four learned models have been tested against the remaining 160 traces. For each model and for each trace s , the probability of generating s has been computed using the forward-backward algorithm [Rabiner, 1989]. Then, s has been assigned to the model with the highest probability. The results reported only one commission error and two rejection errors (no decision taken), when a trace was not recognized by any one of the models.

4 Text Typing Model

The second experiment addressed the more general problem of modeling a user during a text editing activity. A corpus of several paragraphs, selected from newspapers and books, has been collected. The total number of words was 2280, and the number of typed keys 14273. Again, four users typed the entire corpus in several different sessions, without any constraint, in order not to modify their natural typing style. In this kind of application, a user model should be centered not on

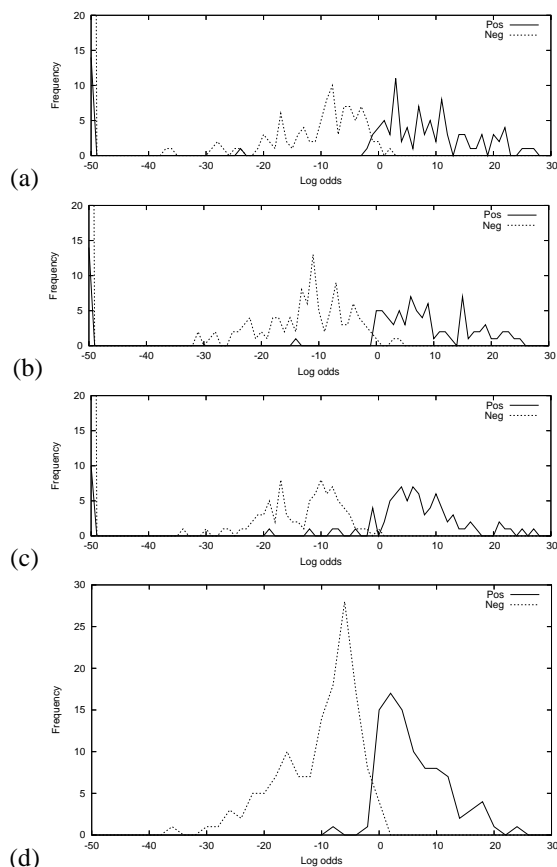


Figure 1: User profiling statistics. Graphs (a), (b), (c) and (d) refer each one to a different user profile. The continuous line "Pos" reports the scoring, measured in log odds, for the sequences belonging to the profile. The dotted line "Neg" refers to the sequences not belonging to the profile.

the specific words he/she types, but on the user typing style, which, in turns, depends on the position of the keys on the keyboard. Therefore, a standard keyboard subdivision into regions, used in dactylography, has been considered. On this basis, keys have been grouped into 10 classes. In this way, transition from one region of the keyboard to another should be emphasized. Afterwards, the sequences generated during a typing session have been rewritten by replacing every character with the name of the class it has been assigned to. Finally, long sequences deriving from an editing session have been segmented into shorter sequences, setting the breakpoint in correspondence of long gaps. The idea is that typical delays due to the user typing style cannot go beyond a given limit. Longer delays are imputable to different reasons, such as thinking or changing of the focus of attention. In this way a set of about 1350 subsequences has been obtained. For every user, a subset of 220 subsequences has been extracted in order to learn the corresponding model. The remaining ones have been used for testing. As in the previous case, the probability of generating each one of the sequences in the test set has been computed for every model. The results are summa-

rized in Figure 1, where the distribution of the scoring rate on the test sequences is reported for every model. The scoring rated is measured in $\log odds$ ¹. The continuous line, labelled "Pos", represents the distribution of the scores assigned to the correct model (user), whereas the other one, labelled "Neg", represents the distribution of the scores assigned to all other models (users), considered together. The sequences on the extreme left have been rejected. It is evident from the figure that sequences belonging to the model are well separated from the other ones. Referring to the data in the test set, a monitoring system using the simple rule that, in a set of three consecutive sequences generated by a user at least two must have a score higher than '0', would give a perfect discrimination of the legal user without rising false alarms.

It is worth noting that the results have been obtained as a first shot, without requiring any tuning of the algorithm. This means that the method is robust and easy to apply to this kind of problems.

5 Conclusion

An improved version of the algorithm [Botta *et al.*, 2004] for inferring complex HHMM from sequences has been applied to a non trivial user profiling problem. Even if this case study is just a preliminary investigation, the results are promising. In fact, the considered case is highly representative of many other similar problems found in intrusion detection systems, and the results show that HHMM is a suitable tool for building profiles. The methodology is effective, robust and easy to apply.

References

- [Bleha *et al.*, 1990] S. Bleha, C. Slivinsky, and B. Hussein. Computer-access security systems using keystroke dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(12):1217–1222, 1990.
- [Botta *et al.*, 2004] M. Botta, U. Galassi, and A. Giordana. Learning complex and sparse events in long sequences. In *Proceedings of the European Conference on Artificial Intelligence, ECAI-04*, Valencia, Spain, August 2004.
- [Brown and Rogers, 1993] M. Brown and S.J. Rogers. User identification via keystroke characteristics of typed names using neural networks. *International Journal of Man-Machine Studies*, 39:999–1014, 1993.
- [Fine *et al.*, 1998] S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine Learning*, 32:41–62, 1998.
- [Lee *et al.*, 2002] W. Lee, w. Fan, M. Miller, S.J. Stolfo, and E. Zadok. Toward cost-sensitive modeling for intrusion detection and response. *Journal of Computer Security*, 10:5–22, 2002.
- [Rabiner, 1989] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of IEEE*, 77(2):257–286, 1989.

¹The logarithm of the ratio between the probability that the observed sequence is generated by the model and the probability that it is generated by a random process.