Using Statistico-Relational Model for Activity Recognition in Smart Home

Alexis Brenon[1], François Portet[1], and Michel Vacher[2] * **

[1] Univ. Grenoble Alpes, LIG, F-38000 Grenoble, France
[2] CNRS, LIG, F-38000 Grenoble, France
{alexis.brenon, francois.portet, michel.vacher}@imag.fr

Abstract. This paper presents the use of a model which mixes logical knowledge and statistical inference to recognize Activities of Daily Living (ADL) from sensors in a smart home. This model called Markov Logic Network (MLN) has different implementations and we propose to compare three of them, from the widely used Alchemy to the new generic framework DeepDive. Finally, we discuss the interest these software products can have for real time activity recognition.

Keywords: Activity Recognition, Markov Logic Network, Factor Graph, Smart Home

1 Introduction

Smart home, as described by De Silva [1], is a home-like environment (home, flat, etc.) equipped with sensors and actuators. These devices can be used to provide various facilities to the inhabitants through ambient intelligence and automatic controls. As listed by Peetoom *et al.* in their literature review [2], many laboratories have built their own smart home to study how technologies can make life easier for people. Among those, researchers are interested in identifying Activities of Daily Living (ADL) to characterize the user's context.

The *context* is a generic term which generally regroup the location of the user, its current activity, etc. [3,4] It is used by context-aware services, and so, activity recognition is crucial to allow context-aware applications to provide more active services.

Activity recognition has been studied for a while and many techniques have been used. One of them relies on computer vision which leads to ethical, acceptance and computational problems. To get around this issue, many projects use only simple and ubiquitous sensors [2] and more recently, a project used microphones to extract information from the audio stream [5].

Given all these kinds of data, inference have been done following two main approaches. On the one hand, a set of rules can be defined by a domain expert to

* This work is part of the CASSIE project founded by a Projet Investissement d'Avenir (2014-2018)

infer high-level information from low-level sensors' data [6]. On the other hand, machine learning and statistical models learned on corpus can be used to generalize already seen behaviors [7]. A third approach, explained in Section 2, unifies the logical models and the statistical ones [8].

Main implementations of statistico-relational model will be compared in Section 3 against a multi-modal corpus. Finally, we will discuss the different perspectives considered in Section 4.

2 Statistico-Relational Models

Markov Logic Network (MLN) is one of the statistico-relational models, primarily introduced by Domingos *et al.* [8]. It is a template model to construct Markov networks (also named Markov random fields) from a set of weighted first-order logic formulas. Based on both of these pillars, MLN combines their advantages to handle at the same time the complexity and the uncertainty of a representation of the real world. As Domingos *et al.* explain in their paper, the MLN formalism subsumes many other probabilistic models allowing it to be more concise and to handle easily non-independent and identically distributed models.

MLN is used to infer the most probable state of a world given some evidences which is called Maximum A posteriori Probability/Most-Probable Explanation (MAP/MPE) inference. This can be easily done using a *SAT* solver able to handle the weights of the formulas [8]. Another task of MLN is to compute the probability that a formula is true in a specific world. This is done thanks to the Markov Chain Monte Carlo inference algorithm.

With the logical models, a domain expert is needed to transfer its knowledge. MLNs take a step to avoid this expensive phase, allowing it to learn the weights from an annotated corpus. In the same way, some attempts have been made to learn the structure of a MLN using Inductive Logic Programming techniques.

It exists different implementations of MLN. The most well known is Alchemy[3] [8] developed by the University of Washington. It makes it possible to define MLNs based on a defined syntax similar to Prolog and to proceed to weight learning and inference on given data. Developed in C++, Alchemy is, to best of our knowledge, no longer maintained since 2013. Nevertheless, it is still widely used by researchers, as in the previous study made in our team. That is why we decided to use it as a baseline to compare other implementations.

Then, Stanford University launched another project, called Tuffy[4], using a similar implementation to Alchemy moving from C++ to Java and relying on PostgreSQL. Usage of an external relational database management system (RDBMS) improves dramatically the time and space efficiency of Tuffy inference phase [9]. In 2014, Tuffy development stopped in favor of DeepDive.

[3] `http://alchemy.cs.washington.edu/`
[4] `http://i.stanford.edu/hazy/tuffy/`

DeepDive[5] is the new project of the Hazy Research group[6]. It is a generic framework primarily designed for knowledge base creation (KBC) which can handle many problems [10]. Born out of the ashes of Tuffy, DeepDive also use the external RDBMS PostgreSQL to manage the data. But DeepDive does not use the exact MLN definition of Domingos *et al.* to handle the logical and statistical parts of the model. Its implementation is based on Factor Graphs (FG) instead of Markov networks. FG is a very generic graphical model which can handle many other problems solved with different graphical models [11].

3 First Results on ADL Recognition

Until now, we have measured the learning performance of the different available software. The learning performance is the score of the system when it infers on the exact same data set than the learning one.

In our case, we use a multi-modal corpus which regroups a full range of sensors values, some computed higher level data (room where the user stayed the longest, agitation level, etc.) and the actual activity of the user. This corpus is composed of 26 hours of experiments on 21 persons, divided into 1 minute long time windows, represented by 94 values [5]. For our first approach, the MLN structure (logical knowledge) is very naive, as we suppose that all the feature values are relevant. Thus, every value implies an activity with a certain degree of truth. This is not optimal but gives us a first baseline to beat.

We ran the following experiments: (1) weight learning and inference with Alchemy; (2) inference with Tuffy (based on Alchemy learning); (3) weight learning and inference with DeepDive. We used Alchemy 1.0, Tuffy-0.3 and DeepDive-0.6.0 on a computer with 24 Intel Xeon CPUs at 2.4GHz with 145GB of RAM running Debian 8 (Jessie) in all of our experiments.

As you can see in Table 1, the execution time of Tuffy and DeepDive is improved by an order of magnitude compared to Alchemy. The small differences in F1-score let's suppose that the three approaches are comparable in performance. We also extract the weights learned by Alchemy and by DeepDive. The correlation between the two sets of weights is about 0.69, highlighting a quite high positive linear correlation which can confirm that both systems work the same way, and that their results can be seen as equivalent.

[5] `http://deepdive.stanford.edu/`
[6] `https://twitter.com/HazyResearch`

	Learning time	Inferring time	F1-Score
Alchemy	+3 days	3h 45'	89 %
Tuffy		3'51"	90 %
DeepDive	23"	10"	91 %

Table 1. Execution time and F1-score of different implementations of statistico-relational models

4 Perspectives

One of the goals of our project is to build a system able to give context information to artificial intelligence. As explained in the Section 1, the user activity is one of the context components.

To do so, our system must be able to infer user activity in *real time* (the user must not feel a lag because of computation time) and to adapt itself to the user, all along its life. Sensor values are coming all along the day, and so our system must be able to handle on-line classification or incremental learning. DeepDive is designed as an incremental framework, and can execute the next learning and inferring phases in less time than the one measured in a one shot test (see Table 1) as shown in its description paper [10].

Then, we will try to implement a reinforcement learning algorithm, decreasing weight of irrelevant formulas. This raises different questions, as in which proportion can we change the weight? Can we change only one weight, or does the modification must impact other weights? How to know which weights to modify?

Finally, we would like the user be allowed to add or remove some formulas without any technical skills. Once these formulas are added to our model, how can we take them into account? Does our model must be fully re-learned?

References

1. De Silva, L., Morikawa, C., Petra, I.: State of the art of smart homes. Engineering Applications of Artificial Intelligence **25**(7) (2012) 1313–1321
2. Peetoom, K.K.B., Lexis, M.A.S., Joore, M., Dirksen, C.D., De Witte, L.P.: Literature review on monitoring technologies and their outcomes in independently living elderly people. Disability and Rehabilitation: Assistive Technology (2014) 1–24
3. Dey, A.K.: Understanding and using context. Personal Ubiquitous Comput. **5**(1) (January 2001) 4–7
4. Coutaz, J., Crowley, J.L., Dobson, S., Garlan, D.: Context is key. Communications of the ACM **48**(3) (2005) 49–53
5. Chahuara, P., Fleury, A., Portet, F., Vacher, M.: On-line human activity recognition from audio and home automation sensors: comparison of sequential and non-sequential models in realistic smart homes. Journal of Ambient Intelligence and Smart Environments (Accepted Paper)
6. Skillen, K.L., Chen, L., Nugent, C.D., Donnelly, M.P., Burns, W., Solheim, I.: Ontological user modelling and semantic rule-based reasoning for personalisation of help-on-demand services in pervasive environments. Future Generation Computer Systems **34**(0) (2014) 97 – 109 Special Section: Distributed Solutions for Ubiquitous Computing and Ambient Intelligence.
7. Cook, D.J., Crandall, A.S., L., T.B., C., K.N.: Casas: A smart home in a box. IEEE Computer **46**(6) (2013) 26–33
8. Domingos, P., Kok, S., Poon, H., Richardson, M., Singla, P.: Unifying logical and statistical ai. In: Proceedings of the Twenty-First National Conference on Artificial Intelligence, AAAI Press (2006) 2–7
9. Niu, F., Ré, C., Doan, A., Shavlik, J.W.: Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. PVLDB **4**(6) (2011) 373–384

10. Wu, S., Zhang, C., Wang, F., Ré, C.: Incremental knowledge base construction using deepdive. CoRR **abs/1502.00731** (2015)
11. Loeliger, H.A.: An introduction to factor graphs. Signal Processing Magazine, IEEE **21**(1) (Jan 2004) 28–41