Introduction to Probabilistic Soft Logic

With a Vision of Using it on Internet of Things Applications

Duhai Alshukaili

Information Technology Department, Ibri College of Technology

Abstract- Classical machine learning approaches make that assumption that data is composed of identically structured objects. However, in many real-world domains such as social networks and data mining, this assumption cannot be made. Statistical relational learning (SRL) is a subfield of machine learning that aims to build probabilistic models of relational data, i.e., data that capture the structure and relationships of objects. This paper reviews an SRL approach, namely Probabilistic Soft Logic (PSL), and then provides a vision of using PSL in IoT applications.

Index Terms- Probabilistic Soft Logic, Statistical Relation Learning, Internet of Things(IoT).

I. Introduction

In many machine learning domains, the assumption that data is composed of identically structured objects cannot be made[1], [2]. Examples of such domains include social networks, the Web, natural language, and so on. To model such domains, it is necessary to capture not only the structure (in the form of attributes) of individual objects but also the relationships between them. For example, to accurately model a social network, one needs to capture the dependencies induced by relationships (e.g., friendship) between individuals in the network. Using relational data in such domains leads to more accurate results from traditional machine learning tasks such as classification and clustering [3]. For example, individuals in a social network are likely to have similar interests depending on whether they are friends or not. Statistical Relational Learning (SRL) is a subfield of machine learning that seeks to build probabilistic models of relational data, i.e., data that capture not just objects bud also relationships between objects[2].

SRL approaches can be categorised by their choice of representation, and by probabilistic semantics for dealing with uncertainty. Representations is often done through either logic (e.g., first-order-logic) or frame-based (e.g., entity-relationship models) representations. Most SRL approaches are based on probabilistic graphical models (PGMs) [4]. Two common types of graphical models used in SRL are Bayesian networks and Markov Networks.

SRL approaches that combine first-order-logic (FOL) and PGMs are suitable for reasoning with uncertainty in the relational domain. In such approaches, weighted FOL rules are defined that model a domain of interest. The model is expressed in terms of first-order predicates and functions. Given some input data, which have been pre-processed as ground *evidence predicates*, the weighted FOL rules then are grounded into a PGM. This grounded PGM provides the probabilistic semantics of the domain and can be used to perform inference over so-called *query predicates*. In this paper, we show how an SRL approach, namely PSL[5], can be used assimilate different facts in a principled way. Different facts can be expressed as logical predicates defined over input data. The dependencies induced by the different kinds of facts are encoded as FOL rules. The uncertainty associated with combination of facts is captured by the parameters of the PGM that results from grounding FOL rules. The combination of FOL and PGMs allows for joint inference, i.e., probabilistic inference in the presence of different sorts of evidence, of claims about objects in the domain.

In this paper, we review PSL and survey a number of problems that have been attempted using a PSL approach. In Section II, we introduce preliminary concepts that pertain to PSL. Section III, provides a vision of using PSL to tackle inference on the Interent of Things (IoT) networks. Finally, Section VI concludes with some remaraks about PSL and our future work.

II. A Brief Introduction Probabilistic Soft Logic

PSL is a relatively recent SRL approach that unified logic and probability. PSL provides a language for encoding expressive domain knowledge through FOL formulae while handling the uncertainty that arises from combining different facts through graphical models, viz., Markov networks. One of the key feature of PSL, when compared with prior approaches such as Markov Logic Networks (MLNs)[1], is that its ground atoms have soft, continuous truth values in the interval [0, 1] rather than binary truth values used in classical logic. In this section we briefly introduce PSL in terms of it syntax and semantics.

A PSL program is a set of weighted first-order-logic formulae in the form $w: A \leftarrow B$, where w are non-negative weights, B is a conjunction of literals and A is a single literal. Consider the following example PSL program (adapted from [6]):

$$0.3: votesFor(B,P) \leftarrow friend(B,A)\ \wedge\ votesFor(B,P) \tag{1}$$

$$0.8: votesFor(B,P) \leftarrow spouse(B,A)\ \wedge\ votesFor(B,P) \tag{2}$$

Intuitively, this PSL program states that given any two individuals a, b and p, instantiation (resp.) the logical variable A, B and P, a claim is made that if b is either a friend or a spouse of a and a votes for party p, then with some likelihood, b votes for p. The respective weights assert that the influence of spouses on what part b votes for is larger than that of friends.

Softness in PSL arises from the fact that truth values are drawn from continues interval $[0, 1]$, i.e., if $\mathcal{A}$ is the set $\{a_1, \dots, a_n\}$ of atoms, then an interpretation is a mapping $I: \mathcal{A} \rightarrow [0,1]^n$, rather than to the extreme values, i.e., either 0 (denoting falsehood) or 1 (denoting truth).

To capture the notion that different claims (expresses as rules) may have different likelihoods, a probability distribution is defined over interpretations, as a result of which rules have more supporting instantiations are more likely. In the case of Rules (1) and (2) above, interpretations where the vote of an individual agrees with the vote of many friends, i.e., satisfies many instantiations of Rule (1) are preferred over those that do not. Moreover, where a tradeoff arises between using agreement with friends or with spouses, the latter is preferred due to higher weight of Rule (2).

Determining the degree to which a ground rule is satisfied from its constituent atoms requires relaxing (with respect to the classical definitions) the semantics of logical connectives for the cases where terms takes soft truth-values. To formalize this, PSL uses the Łukasiewicz t-norm and its corresponding co-norm, which are exact (i.e., coincide with the classical cases) for the extremes and provide a consistent mapping for all other values.

Atom in a PSL program can be user-defined. Thus, unary predicate `IsDictWord` might be defined to have truth value 1 if the individual of which it is predicated is a dictionary word and 0 otherwise. Atoms in PSL can also capture user-defined relationships between sets of individuals. Thus the truth value `SimInterests(S, T)` can be defined as the similarity of the respective sets of interests of the individuals `S` and `T` as computed by some similarity function.

In summary, the basic idea is to view logical formulas as soft constraints on their interpretations. If an interpretation does not satisfy a formula, it is taken as less likely, but not necessarily impossible. Furthermore, the more formulas an interpretation satisfies, the more likely it is. In PSL, this quantification is ground on the relative weight assigned to each formula. The higher the weight, the greater the difference between the likelihood of an interpretation that satisfies a formula and the likelihood of one that does not.

The key tasks supported by PSL implementations are learning and inference. The rules in a PSL program can be either given (i.e., asserted) or learned from training data. Furthermore, rule weights are learned from sample data. The weight learning process takes a PSL program (possibly with initial weights), a specification of both evidence and query predicates, and sample data. A predicate is said to be an evidence predicate if all ground atoms have known truth values by observation. A predicate is said to be a query predicate if one or more of its ground atoms have unknown truth values. The process returns a relative non-negative weight for each rule. A positive weight denotes that a rule is supported by the sample data whereas the magnitude indicate the strength of the support. A weight of zero denotes lack of support in the sample data for that rule.

The main purpose of a PSL program is inference. The PSL inference process takes a PSL program, evidence as data, and a query. It then computes the most probable assignment of soft-truth values to the query, i.e., the probability that a given query atom is true. A major strength of PSL is that implementations perform inference by constructing a corresponding convex optimization problem for which a solution can be efficiently computed event for large-size inputs. For detailed descriptions of PSL learning and inference algorithms, see[5].

III. Inference On The Internet of Thing using PSL

The Internet of Things (IoT) is a buzzword used to refer to an ecosystem that includes communication, applications, data, and analytics. IoT has opened the door for exciting opportunities for novel and challenging applications such as (among others) smart cities, smart buildings, intelligent traffic control, and health monitoring. IoT ecosystem can be viewed as layered architecture[7] depicted in Figure 1.

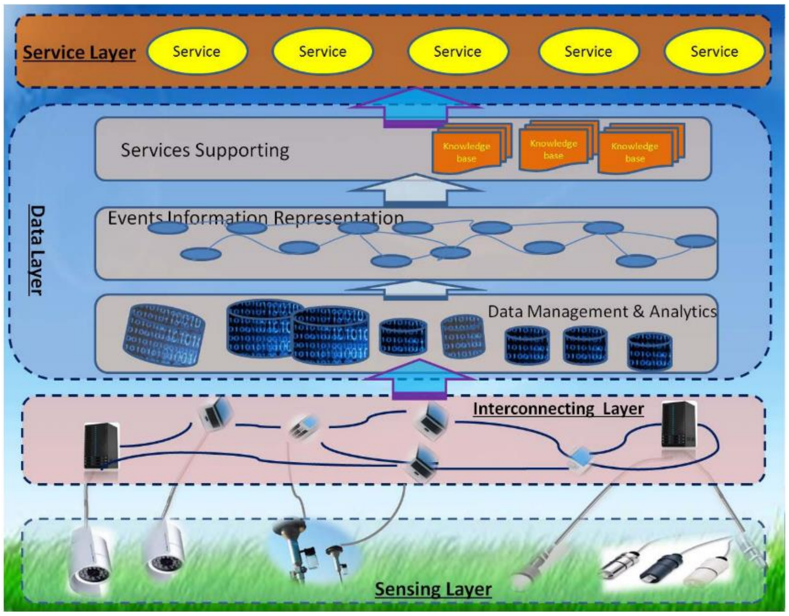

Figure 1:Layered Architecture of the Internet of Things (Adopted from [7]*)*

At the bottom is the sensing layer which main purpose is to collect raw data from the sensing devices and passing across the IoT network infrastructure. Such devices may not work perfectly all the time and their data might be unreliable due to the fact that electronic device may fail over a period of time[8].

The interconnectivity layer handles data transmission across different devices and domain in the IoT network. A study on the security of smart homes IoT networks[9] has shown that an attacker can manipulate the sensor data which increases the uncertainty of the network data.

The data layer is responsible for storage and retrieval of heterogenous data stemming from bottom two layers and extracting useful pattern through data mining and machine learning algorithms. It is known that machine learning algorithms mistakes[10]. However, the performance of automated inference algorithms can be improved through machine learning paradigm known as human-in-the-loop[11], [12]. The main idea of this paradigm is that human interacts with the results of the machine learning system through which can learn better way to improve its results. Such systems has shown some success in areas such as data integration[13], [14].

Therefor one can notice that there is an inherent reliability and uncertainty problems that may arise in IoT systems. Such problems may hinder the trust of the services that rely on such systems. It has been demonstrated that PSL can be useful for encoding uncertain knowledge on given domain and compute inference based on observed events.

Consider for example the application of smart waste management where the goal is to improve the efficiency of waste collection. In order to make a decision on whether to send truck to a certain area within the city depends on a number of events that can be classified as either **simple** or **complex**. A simple is a significant change in the input data. An example of this event could be a reading from some waste bin that is full or near full. A complex event can be a combination of two or more simple events such as the reading from two waste bin that they are full. An IoT application should be capable of responding to both kinds of events. In addition, a domain expert should be able to state different kinds of ways complex events could occur simultaneously.

Because it is based on FOL, using PSL a domain expert can easily state simple and complex events how the system should respond to these events. Consider the following PSL program:

```
1.0:   SendTruck(tid, loc)  ← Bin(bid, loc) ^ isFull(bin)

3.5:   SendTruck(tid, loc) ← Bin(bid1, loc) ^ Bin(bid2, loc) ^ bid1 != bid2 ^
                             isFull(bin1) ^  isFull(bin2)
```

The above PSL program states that if bin is full than send a truck to the location of that bin and if two bin that are located in the same area than and a truck that area. Note that the second rule has a higher weight, therefore an inference system that uses these rules will give more emphasis to location that have bins going full within the same area of the city.

One of the advantages of using PSL to model this problem is that one can easily incorporate different sorts of evidence to make the necessary decisions (i.e., sending a truck to a location in the above example). Such evidence can be stem from sensors, data analytics on raw data, or even human feedback on the system. PSL is an efficient and rubout framework that have the expressive power of FOL and allows for uncertainty management through graphical models. Using PSL will allow for fusing data coming from multiple heterogeneous sources of information to preform inference on IoT environments.

IV. Conclusion and Future Work

With the view of managing the complexity and uncertainty of IoT systems, this paper proposes the utilization an SRL approach called PSL. In future work we envisage to study the use PSL in IoT system in more details by testing PSL models on deployed IoT systems and evaluate it against bespoke data analytics and machine learning approaches.

References

[1] P. Domingos and D. Lowd, “Markov logic: An interface layer for artificial intelligence,” *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–155, 2009.

[2] L. Getoor and B. Taskar, *Introduction to statistical relational learning*. MIT press, 2007.

[3] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, “Collective classification in network data,” *AI Mag.*, vol. 29, no. 3, p. 93, 2008.

[4] D. Koller, N. Friedman, L. Getoor, and B. Taskar, “Graphical Models in a Nutshell,” *Stat. RELATIONAL Learn.*, p. 13, 2007.

[5] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, “Hinge-Loss Markov Random Fields and Probabilistic Soft Logic,” *J. Mach. Learn. Res.*, vol. 18, no. 109, pp. 1–67, 2017.

[6] A. Kimmig, S. Bach, M. Broecheler, B. Huang, and L. Getoor, “A short introduction to probabilistic soft logic,” in *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, 2012, pp. 1–4.

[7] Y. Sun, H. Song, A. J. Jara, and R. Bie, “Internet of Things and Big Data Analytics for Smart and Connected Communities,” *IEEE Access*, vol. 4, pp. 766–773, 2016.

[8] T. Jin, P. Wang, and Q. Huang, “A practical MTBF estimate for PCB design considering component and non-component failures,” in *RAMS’06. Annual Reliability and Maintainability Symposium, 2006.*, 2006, pp. 604–610.

[9] M. R. Schurgot, D. A. Shinberg, and L. G. Greenwald, “Experiments with security and privacy in IoT networks,” in *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015, pp. 1–6.

[10] P. Domingos, “A Few Useful Things to Know About Machine Learning,” *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012.

[11] A. Holzinger, “Interactive machine learning for health informatics: when do we need the human-in-the-loop?,” *Brain Informatics*, vol. 3, no. 2, pp. 119–131, 2016.

[12] D. Xin, L. Ma, J. Liu, S. Macke, S. Song, and A. Parameswaran, “H elix: accelerating human-in-the-loop machine learning,” *Proc. VLDB Endow.*, vol. 11, no. 12, pp. 1958–1961, 2018.

[13] N. W. Paton, K. Belhajjame, S. M. Embury, A. A. A. Fernandes, and R. Maskat, “Pay-as-you-go Data Integration: Experiences and Recurring Themes,” in *SOFSEM 2016: Theory and Practice of Computer Science - 42nd International*

Conference on Current Trends in Theory and Practice of Computer Science, Harrachov, Czech Republic, January 23-28, 2016, Proceedings, 2016, vol. 9587, pp. 81–92.

[14] S. R. Jeffery, M. J. Franklin, and A. Y. Halevy, "Pay-as-you-go user feedback for dataspace systems," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 847–860.

AUTHORS

Correspondence Author – Duhai Alshukaili, PhD in Computer Science, Ibri College of Technology, duhai.alshukaili@ibrict.edu.om.