Indian Journal of Science and Technology, *Vol 10(19), DOI: 10.17485/ijst/2017/v10i19/113384, May 2017*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Recent Advances in Markov Logic Networks

Romany F. Mansour[1] and Samar Hosni[2]

[1]Faculty of Science, New Valley - Assiut University, Egypt; romanyf@aun.edu.eg
[2]Computer and Information Technology College, Northern Border University, KSA; samar_hosni@hotmail.com

Abstract

Objectives: To identify recent progress and areas of application for one technique in soft computing, specifically. This technique is known as Markov Logic Networks. **Methods/Statistical Analysis:** Soft computing combines machine learning and fuzzy logic in order to tackle problems that appear to have no definite solution. In doing so, soft computing approaches a human style of thought, and lends itself well to data-rich, heterogeneous and fast-changing scenarios. The success of soft computing has only fueled to drive for better, more powerful, and faster algorithms. **Findings:** Soft computing has already revolutionized a number of fields, including artificial intelligence, robotics, voice recognition, and areas of biomedicine. It has the potential to continue doing so, but this future success depends heavily on making more ambitious soft-computing algorithms tractable and scalable to Big Data – sized problems. One promising technique that has come to the forefront of soft computing research in recent years is the heavily probabilistic-reasoning-based Markov Logic Network (MLNs). MLNs combine the efficiency of the Markov Model with the power of first-order logical reasoning. MLNs have already proven themselves adept at such futuristic implementation as smart homes, voice recognition, situations awareness, prediction of marine phenomena, and weather assessment. In order to make MLNs more tractable, research has recently turned towards normalizing progressively by time-slice to assure convergence, and "lifting" structural motifs from similar, already-computed networks. Progressive efforts in these areas should deliver a next-generation of situation awareness in "smart" electronics and predictive tools, one more step towards true artificial intelligence. **Application/Improvements:** Soft computing has already revolutionized a number of fields, including artificial intelligence, robotics, voice recognition, and areas of biomedicine. It has the potential to continue doing so.

Keywords: Evolutionary Algorithms, Fuzzy Logic, Machine Learning, Markov Logic, Soft Computing

1. Introduction

Soft computing is an area of computer science that aims to solve difficult, complex problems that are not tractable by usual, deterministic computing approaches. Soft computing is tolerant of imprecision, partial truth, and rough estimates or approximations. Some proponents of soft computing argue that it is similar to processes employed by the human mind itself. It can also be said that soft computing aims to solve sets of problems that are NP-complete, or problems that are solvable on the order of a natural polynomial function with respect to time. This is an important distinction, as problems in Big Data and science may often extend to an exponential level of complexity of an exact, deterministic or "hard" solution is sought[1]. Soft computing depends upon a collection of other areas of computation and mathematical reasoning. These include fuzzy logic, machine learning, and probabilistic reasoning[2]. Often, interdisciplinary and/or cutting edge new fields in computation or artificial intelligence rely on disciplines that are redundant or at least competing. Soft computing, by contrast, enjoys the unusual disposition of depending on a set of complementary subfields[3]. Fuzzy logic, machine learning, and probabilistic reasoning have different strengths and weaknesses, and therefore are best applied to different areas of soft computing. Often, all three are required in a unified framework to properly "solve" (i.e. approximate within the required precision or above the required level of performance) the soft computing problem.

**Author for correspondence*

It would be incorrect to view soft computing as only being able to handle tasks that are less important than those that are amenable to deterministic, "hard" computing solutions. On the contrary, generally soft computing aims to solve problems that would be impossible by any other method. In fact, soft computing is often viewed as approaching conceptual intelligence, or generalized intelligence with respect to a particular environment or ontological domain[4]. Examples include intelligent search algorithms for the world-wide web, biomedical inference, robotics, and smart devices / smart environments. Soft computing will become increasingly important in coming years in science and engineering. Ultimately, soft computing may approximate something like the human mind's ability to store and process ambiguous, vague, and non-categorical information. Already, a Machine-Intelligence Quotient (MIQ) metric has been developed[5]. This metric has been used to measure the effectiveness and, equally importantly the situation and environmental awareness of computer devices and algorithms. Institutes such as the Berkeley Initiative on Soft Computing (BISC) have arisen in order to advance soft computing and increase its range of applications[6].

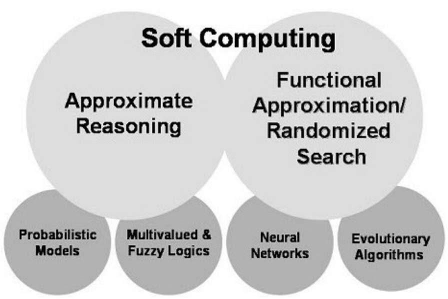

Figure 1. Soft computing is a synthesis of 3 complementary disciplines. These 3 disciplines are: Fuzzy Logic (FL), Machine Learning (ML) (combining neural networks and evolutionary algorithms), and Probabilistic Models (PM).

1.1 Fuzzy Logic

Fuzzy logic is a form of approximated logic that deals with partial truths or truths that can be approximated on a scale from 0 to 1, rather than a binary either 0 or 1[7]. This appears to be particularly useful for linguistic reasoning, such as in the case of "hedging", when words or other terms / phrases are qualified by their surrounding of words. Individual functions may be generated for each type of linguistic variable and used to weight the different meanings of the word or phrase in question[8]. It is not because of an inherent limitation in the capabilities of more absolute binary logic – based methods, but rather because of a more pragmatic limitation to such methods, that binary logic methods are so useful[9]. The data that scientists and engineers employing soft computing techniques generally work with are often of an "organic" type; especially of Big Data applications are being implemented. To be more precise, sets of data are often poorly organized and thus overlap to a considerable extent. It is the ability to deal with unclear boundaries between datasets and categories that fuzzy, or non-categorical, logic is useful as shown in Figure 1.

1.2 Machine Learning

Machine learning is a branch of artificial intelligence researching that has resulted from decades of work on pattern recognition and computer science. Machine learning focuses on the development of algorithms, or fixed sequences of rules implemented over a data space, that analyze data and make predictions based on significant patterns either discovered from an external template (supervised training) or from structures discovered within the data itself. Machine learning developed, at least in the early phases, in parallel with the field of computational statistics, which deals with methods for inferring significant parameters in observed data and which also focuses on making predictions. Machine learning methods seek to optimize the parameters of the particular model for the task of successful predictions (many parameters may be used for this optimization task, including but not limited to the accuracy of predictions, the specificity, the sensitivity, or the area under the receiver-operator curve which balances specificity with sensitivity)[10]. Because of the usefulness of machine learning in finding important patterns in datasets and using them for the task of making additional predictions, machine learning is essential for data mining and other Big Data tasks, although it is important to keep the distinction between machine learning and these fields.

Applications of machine learning are nearly as diverse as the source of data generated by science and engineering, as well as humanities, disciplines. Machine learning is important wherever computers are intended to act with some degree of independence, i.e. without being specifically programmed for the exact task and dataset in question. Machine learning has recently led to a number of potentially paradigm-shifting technologies, although for the most part these technologies are not yet fully implemented and their potential has yet to be fully realized. These fields include self-driving cars[11], speech recognition[12], understanding of human neural circuits[13,14], robotics[15], and topics in Bioinformatics that are too numerous to list in the present study.

1.3 Probabilistic Reasoning

Probabilistic reasoning, often confused with fuzzy logic, deals with the modeling of a systems from example inputs in order to infer a most likely output or conclusion scenario. Like fuzzy logic, probabilistic reasoning seeks to "escape" from the mold of purely static or binary logical conditions, and instead allow a more graded spectrum of weights for modeling relationships between variables and ultimately generating an output.

Unlike fuzzy logic, probabilistic reasoning deals with rational propositions upon the dataset[16]. These may be of a linguistic or similar human-level set of relationships. Whereas fuzzy logic aims to exploit the overlap between sets of data, probabilistic reasoning aims to exploit the information inherent in uncertain relationships between data objects. Thus, probabilistic reasoning aims to "save" deductive logic by providing logical inferences where uncertain logic relationships exist. It is often said that the human brain works using probabilistic reasoning. One important limitation of probabilistic-reasoning based approaches is that they are difficult to render tractable, although advances towards this end have been made in recent years[17].

1.4 Chaos Theory

Although generally not considered as a primary area of research for soft computing techniques, chaos theory does deal with phenomena that are often associated with the datasets or more specifically imperfections and "noise" in the datasets, that soft computing is often applied to. Chaos theory deals with the study of systems for which the outcomes are highly-sensitive to the initial conditions[18]. In particular, chaos theory deals with such systems when they are not amenable to analysis and modeling by usual, deterministic methods[19]. Despite the fact that such systems lead to extremely complex mathematical and computational problems, they can often be the result of simple inputs. For instance, a billiard ball table is likely to have a very different state depending on whether a ball is struck at one angle or an angle very slightly different. Another example is the double-pendulum example, wherein two lengths of a rod are attached by a socket joint. The arc traced by the end of the second problem can be described as a chaotic phenomenon.

Although many chaotic phenomena are difficult to model or approximate, underlying parameters may be inferred whose change over the same time course may be easier to comprehend, e.g. to visualize. A first recurrence map, also known as a Poincare map (named after Henri Poincare) is the path traced by such underlying parameters or state space of the systems for one full course of activity, until they return to their original values (hence "recurrence"). It can be proved that a Poincare map preserves a number of properties and characteristics of the original data space. For example, a Poincare map of the orbit of stars in a galaxy can be used to infer the forces of gravitational pull between the stars and the mass center of the galaxy, and hence the formula for an ellipse.

1.5 Soft Computing and NP-Completeness

As indicated, soft computing is the use of inexact or approximate solutions to tackle challenging problems that would otherwise be intractable. To be specific, soft computing is able to render problems solvable in an NP-complete operational order o complexity. NP-complete means that these problems can be solved in a timeframe equal to the output of some polynomial function of the time. Often, especially with "organic" data increasingly being generated by fields in the life sciences and engineering, the order of operation for solving the problem is closer to an exponential function. This is a natural consequence of the situation wherein each solutions paths are constantly diverging, and all resulting candidate solutions much be checked against one another to verify which is optimal. A deterministic method would be required to explore the entire state space of solutions, and thus after each iteration of solutions, a new iteration would be based on the previous number of existing solutions.

By contrast, with soft computing, approximation allows for entire branches of solutions to be abandoned or collapsed into other, similar lines of problem-solving. This ability to "blur the lines" between paths of solving, or applying "fuzzy logic", allows for great improvements in speed and practicality of the methods. Complex systems requiring the use of soft computing include problems in biology, medical sciences, social science, and data analytics. Most of the avenues for allowing approximation rely in one way or another upon the use of inductive, rather than deductive, reasoning. Inductive reasoning is the process of logical deduction using likelihood estimation rather than absolute proofs. Because inductive reasoning is based on estimations of likelihood, it allows a clear path for moving from general statements to individual case scenarios and hence is often founded more concretely in statistical rigor than pure deductively-reasoned statements.

1.6 Specific Aims

The specific aim of the present study is to identify recent progress and areas of application for one technique in soft computing, specifically. This technique is known as Markov Logic Networks (MLNs). Markov Models encapsulate a set of states, each tied to the other with a specific probability for transitioning. Thus, at any new time instance t+1 following t, the probability of the next state can be approximated exactly with the transition probability. Each such state (referred to as a hidden state), also has its own set of emissions probabilities, or probabilities for producing an external, observed phenomenon. Thus, the observations can be used to optimize a concise, compact formulation for transition probabilities among a collection of hidden states, accompanied each by emissions probabilities for observed states. Markov logic networks are Markov Models that use logical functions or relationships (often containing some form of machine-learning or probabilistic-reasoning – based model) to generate transition and emissions probabilities. Most commonly, MLNs employ first-order logic or general logical propositions involving objects and their relationships (otherwise known as "worlds", given the origin of first-order logic in summarizing declarative linguistic statements containing familiar grammatical objects and syntax). The "world" of objects and their relationships are boiled down to underlying or causative objects (the grammatical "subjects"), observed or resulting objects (the grammatical "objects"), a set of probabilities for transitioning from each subject to the other, and a set of probabilities for each subject to generate "action" or invoke a relationship with respect to each object.

2. Methods

Articles referring to Markov Logic Networks were searched for using peer-reviewed article databases, including Web of Science and Google Scholar. For the initial round of searching, only articles within the last 5 years (since 2010) were considered. Articles predating 2010 were considered if they were found the be a common reference of more than 1 article obtained in the initial search, and thus could be considered as seminal works for the specific area of MLNs. Articles were read, analyzed, and compared to distill a set of primary research directions, themes and computational techniques, and overall methodologies. Areas of application were recorded.

3. Results

The original Markov Logic Network (MLN) was developed by [20]. This network was created in order to represent a first-order knowledgebase using formulas (or clauses) to attach weights. Inference in this prototypical MLN was performed using Markov-Chain Monte-Carlo, or heuristic sampling over a subset of initial conditions until a maximal probability of clauses result (i.e. until the correct first-order logical statement could be refined). This innovation was a major step forward in soft computing, as it represented the marriage of a highly-efficient generative computational framework with an object representational structure of sufficient complexity to model human thought and speech. Since the development of MLNs, they have therefore often been applied to tasks requiring a human level of awareness, such as speech recognition, voice-based instruction, and awareness of human environments. Examples of these cases are elaborated further, below. An illustration of Hidden Markov Models (HMMs) is provided in Figure 2, The MLN is a form of HMM in which the transition and emission probabilities are derived from probabilistic reasoning applied to first-order logic "worlds".

3.1 Areas of application for Markov Logic Networks (MLNs)

Since their original development 9 years ago[20], MLNs have been applied to a variety of complex, human-level tasks. For example MLNs have been developed for the recognition of dementia-type activity in nursing homes. Healthcare systems in smart environments have employed MLNs to screen patients for signs of the onset, or worsening, of dementia, using visual and auditory clues from surveillance devices[21]. The indicated model augmented the native logic of the MLN with "expert" knowledge or common sense modules. The suitability of MLNs for the task of identifying dementia results from the nature of the affliction, which presents as abnormalities in the type of objects, the time, location, and duration of activity with regard to such objects. Maritime environments have also proved themselves amenable to analysis by MLNs. Because a large amount of the typical maritime environment is hidden from plain view (being underwater), the maritime environment is a natural fit for the kind of soft computing that MLNs allow, with their hidden states. Hidden states correspond to unseen underwater conditions, and emitted states correspond to observed effects (wave size/frequency, undercurrent, hue, turbulence, etc.)[22]. Again, an advantage of using MLNs (as opposed to a traditional and simpler HMM) is that the resulting most probable set of states and transitions correspond to interpretable real-world scenarios.

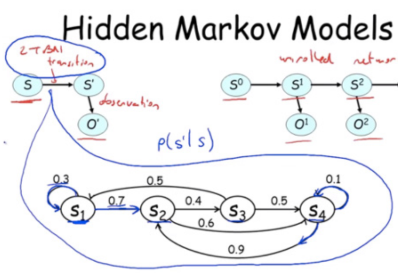

Figure 2. Overview of Hidden Markov Models (HMMs).

One fascinating and futuristic application of MLNs is for providing core speech recognition and environmental awareness in interactive or "smart" homes. If a user of such a home in the future gives the instruction "turn on the lights", the response of the house is clearly dependent upon a variety of different environmental factors, such as whether it is night or day, for example. In the former case, illuminating a bedside lamp may be the appropriate response, but this is clearly insufficient in the latter case. [23,24] show that MLNs can be used for the recognition of different types of activity within the house, giving the house the ability to perform concrete, first-order logical induction and respond accordingly. Smart homes of the future could be populated with pet robots. Already, autonomous robots are performing an increasingly variety of tasks, from street clean-up, to house clean-up, to driving, and ultimately even as retail clerks. Interacting with humans requires the ability to process human speech, the ability to process the environment and generate a logical awareness about it, and finally the ability to synthesize a response to this environment given the original spoken instructions. Instructing a house robot to "clean up" or "set the table" is likely to require a highly-sophisticated analysis and response system in the robot[25]. The system described by [25]proposes a kind of "virtual knowledge base" wherein collections of knowledge pieces are not stored, but rather created "on the fly" by the robot's internal data structures, perception system, and data from external sources.

3.2 Recent advances in Markov Logic Network Design

One of the biggest problems with Markov Logic Networks (MLNs) and with Markov models in general is the problem is low residuals. In order to arrive at the most probable path of state transitions through the Markov Model, with respect to the observations, it is often necessary to multiple a great numbers of very small probabilities. Thus, the final probability becomes extremely small. This causes a problem for a number of reasons, not least that many computers round down infinitesimally small float or double data types to zero. In addition, one faces the problem of small residuals – differences in probability between two different paths become extremely large as a proportion of the total probability of the smaller one, resulting in chaotic convergence behavior (i.e. "noise" leading to infinite loops at the later stages of the convergence algorithm). In addition, the collection of all paths that the optimization routine runs through often are divergent, i.e. they do not all sum up to 1.

This has to do with the time slice problem or the fact that a time slice of defined length is used to increment the algorithm (data collection, recalculating of transition and emissions probabilities, etc.). While the problem of small residuals remains largely unsolved, recent research has advanced a possible solution for the problem of divergent residuals. This problem is exacerbated by the fact that the marginal probabilities of truth assignments can change if the domains of first-order logic predicates (previously-introduced subjects or objects) is altered, either by extending it or reducing it.[26] Proposes a modification the MLNs that fix this problem by normalizing the MLNs across each time "slice". In brief, this simply means that all of the existing residuals are normalized such that their sum is, in fact, 1. This is accomplished not simply by up- or down-weighting all of the probabilities, but by creating an internal Markov Logic Network to model influences between variables that do not have a direct causal effect on each other. This approach appears to be tractable and scalable to online applications.

Another problem with MLNs is that they are so time-consuming, despite being NP-complete, as to make them impractical for a number of applications. The main limitation is scalability. One solution for speeding up MLNs is to "lift", or borrow similar MLN states at different times during the running of the MLN. Properties of a calculated MLN may be generally similar, if the overall states and state relationships are very similar. One remaining limitation of MLN state "lifting" is that the number of possible matches can balloon quite dramatically, making the situation even worse rather than better shown in Figure 3.

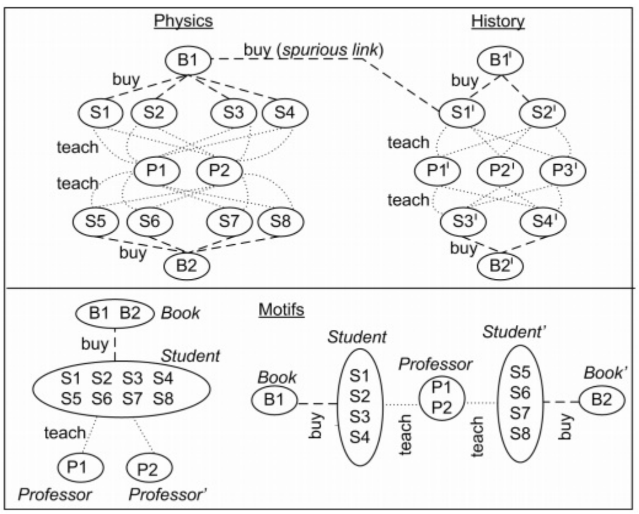

Figure 3. Lifting Structural Motifs from Similar Markov Logic Networks (MLNs).

In[27] Motifs extracted from ground hyper graphs (unrolled MLN, top) for History and Physics classes, involving book, student, and professor objects linked probabilistically by actions (buy, teach) appear to be rather similar (bottom). Thus, transitional probabilities between the two MLNs are likely to be rather similar and one can be lifted from the other (not shown). Thus, there remains an issue of granularity of predictions, since even with normalization by time slice and lifting of structural motifs trade-offs are still necessary in order to maintain tractability with the complex, rich probabilistic reasoning routines used in transition probability / emissions probability calculation. One possible solution has to do with coarse-to-fine grain lifting, or lifting of major structural features initially, followed by lifting of finer features later on[17]. This prevents the problem of ballooning complexity of lifted structural motifs, and allows the MLN to become convergent[28]. Other approaches towards making lifting tractable include lifting by symmetry (this would help accelerate the earlier, coarse-grained stages of the aforementioned coarse-to-fine-grained approach). Finally, certain domains of first-order logical semantics can be orderly into hierarchical relationships, such that the searches for possible "worlds" can more easily converge, i.e. by moving down the tree form general principles and semantic relationships to more specific ones. This effort has led to the development of a so-called "tractable Markov language", or TML.

4. Conclusion

Soft computing has already revolutionized a number of fields, including artificial intelligence, robotics, voice recognition, and areas of biomedicine. It has the potential to continue doing so, but this future success depends heavily on making more ambitious soft-computing algorithms tractable and scalable to Big Data – sized problems. One promising technique that has come to the forefront of soft computing research in recent years is the heavily probabilistic-reasoning-based Markov Logic Network (MLNs). MLNs combine the efficiency of the Markov Model with the power of first-order logical reasoning. MLNs have already proven themselves adept at such futuristic implementation as smart homes, voice recognition, situations awareness, prediction of marine phenomena, and weather assessment. In order to make MLNs more tractable, research has recently turned towards normalizing progressively by time-slice to assure convergence, and "lifting" structural motifs from similar, already-computed networks. Progressive efforts in these areas should deliver a next-generation of situation awareness in "smart" electronics and predictive tools, one more step towards true artificial intelligence.

5. References

1. Bonissone PP. Soft computing: the convergence of emerging reasoning technologies. Soft computing. 1997; 1(1):6-18. Crossref
2. Zadeh LA. Fuzzy logic, neural networks, and soft computing. Communications of the ACM. 1994; 37(3):77-84. Crossref
3. Zadeh LA. Discussion: Probability theory and fuzzy logic are complementary rather than competitive. Technometrics. 1995; 37(3):271-76. Crossref

4. Yao Y. Perspectives of granular computing. 2005: IEEE International Conference on Granular Computing. IEEE. 2005; 1: 85-90. Crossref
5. Park HJ, Kim BK, Lim KY. Measuring the machine intelligence quotient (MIQ) of human-machine cooperative systems. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans. 2001; 31(2):89-96.
6. Zadeh LA. BISC: The Berkeley Initiative in Soft Computing. 2009; 44:693.
7. Klir G, Yuan B. Fuzzy sets and fuzzy logic. New Jersey: Prentice Hall. 1995; 4:1-88.
8. Ganter V, Strube M. Finding hedges by chasing weasels: Hedge detection using Wikipedia tags and shallow linguistic features. Proceedings of the ACL-IJCNLP 2009 Conference Short Papers. Association for Computational Linguistics. 2009 August; p. 173-76. Crossref
9. Zadeh LA. Is there a need for fuzzy logic? Information sciences. 2008; 178(13); 2751-79. Crossref
10. Alpaydin E. Introduction to machine learning. MIT press. 2014; p. 1-579.
11. Markoff J. Google Lobbies Nevada to Allow Self-Driving Cars. The New York Times. 2011; P. 10.
12. Anusuya MA, Katti SK. 2010: Speech recognition by machine, a review. arXiv preprint arXiv. 2010; 6(3):1-25.
13. Fiser J, Berkes P, Orbán G, Lengyel M. Statistically optimal perception and learning: from behavior to neural representations, Trends in cognitive sciences. 2010; 14(3):119-30. Crossref PMid:20153683 PMCid:PMC2939867
14. Modayil J, White A, Sutton RS. Multi-timescale nexting in a reinforcement learning robot, Adaptive Behavior. 2014; 22(2):146-60. Crossref
15. Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann. 2014; p. 1-2.
16. Domingos P, Web WA. A Tractable First-Order Probabilistic Logic. In AAAI. 2012; p. 1-8.
17. Robertson R, Combs A. Chaos theory in psychology and the life sciences. Psychology Press. 2014; p. 416.
18. Zelinka I, Celikovsky S, Richter H, Chen G. Evolutionary algorithms and chaotic systems. Springer Science and Business Media. 2010; 267:560
19. Richardson M, Domingos P. Markov logic networks. Machine learning. 2006; 62(1-2):107-36. Crossref
20. Gayathri KS, Elias S, Ravindran B. Hierarchical activity recognition for dementia care using markov logic network, Personal and Ubiquitous Computing. 2014; p. 1-15.
21. Snidaro L, Visentini I, Bryan K. Fusing uncertain knowledge and evidence for maritime situational awareness via Markov Logic Networks, Information Fusion. 2015; 21:159-72. Crossref
22. Chahuara P, Fleury A, Portet F, Vacher M. Using markov logic network for on-line activity recognition from non-visual home automation sensors. Springer Berlin Heidelberg: Ambient intelligence. 2012; p. 177-92. Crossref
23. Chahuara P, Portet F, Vacher M. Context aware decision system in a smart home: knowledge representation and decision making using uncertain contextual information. The 4th International Workshop on Acquisition, Representation and Reasoning with Contextualized Knowledge (ARCOE-12). 2012; p. 52-64.
24. Tenorth M, Beetz M. KnowRob: A knowledge processing infrastructure for cognition-enabled robots, The International Journal of Robotics Research. 2013; 32(5):566-90. Crossref
25. Papai T, Kautz H, Stefankovic D. Slice normalized dynamic markov logic networks. In Advances in Neural Information Processing Systems. 2012; p. 1907-15.
26. Kiddon C, Domingos P. AAAI: Coarse-to-Fine Inference and Learning for First-Order Probabilistic Models. 2011; p. 1-8.
27. Kok S, Domingos P. Learning Markov logic networks using structural motifs. Proceedings of the 27th International Conference on Machine Learning (ICML-10). 2010; p. 551-58.
28. Ahmadi B, Kersting K, Mladenov M, Natarajan S. Exploiting symmetries for scaling loopy belief propagation and relational training, Machine learning. 2013; 92(1):91-132. Crossref