Markov Logic Article Review

Juan Antonio Bernabé Moreno

Abstract— **Starting with a brief summary of the Markov network's article authored by Pedro Domingos, this work subjects it to a strengths and weaknesses analysis. After that, the bibliographic references are checked.**
The final part focuses on an outlook to newer works published in these research areas.

I. Summary

Pedro starts this work introducing the two core challenges of the machinery learning and the frameworks to address them: probability for dealing with uncertainty and first-order logic for dealing with complexity.

After this introduction the goal setting statement comes: a combined approach where machine learning and inference is possible. Following the goal setting, a briefly review of historical approaches up to the moment that combine a probabilistic graphical model with a subset of first-order logic: stochastic logic programs, probabilistic relational models, Bayesian logic programs, relational dependency networks.
The Markov logic is introduced as a model with two key-features:

1. Conceptually simple providing the graphical models' full expressiveness
2. Remains well-defined in many infinite domains.

It is the result of extending the 1st-order logic with the attachment of weights to formulas that are viewed as template for constructing Markov networks. Should weights be infinite, then the Markov logic would be equivalent to 1st-order logic [4].
Different from statistical learners, the Markov logic doesn't assume that data are independent and identically distributed. It leverages the power of 1st order logic to represent dependencies between objects.

In order to explain the Markov logic model, the author starts describing the two components separately:

A. Markov Network

Is defined as a model for the joint distribution of a set of variables

$$X = (X_1, X_2, \ldots, X_n) \in \mathcal{X}$$

Undirected graphical models and set of potential functions
The potential functions are defined over cliques:

$$P(x) = \frac{1}{Z}\prod_c \Phi_c(x_c)$$

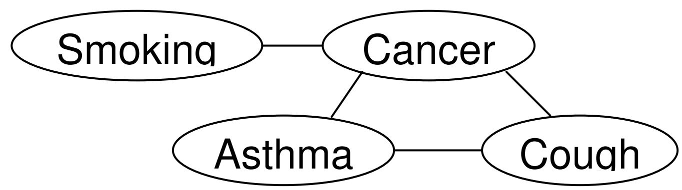
Where x_c is the state of the *kth* clique. Z is known as the partition function, given by:

$$Z = \sum_x \prod_c \Phi_c(x_c)$$

For example for the objects Smoking and cancer the potential functions look like:

Smoking	Cancer	Φ(S,C)
False	False	4.5
False	True	4.5
True	False	2.7
True	True	4.5

And the undirected graphical model as follows:



A Markow network can also be represented as a *log-linear model*:

$$P(x) = \frac{1}{Z}\exp\left(\sum_i w_i f_i(x)\right)$$

The $f_i(x)$ is called feature and w_i the weight for this feature
Continuing with the smoking sample:

$$f_1(\text{Smoking, Cancer}) = \begin{cases} 1 & \text{if } \neg \text{ Smoking} \vee \text{Cancer} \\ 0 & \text{otherwise} \end{cases}$$

$$w_1 = 1.5$$

B. First-Order logic

Pedro introduces the concept of 1st-order knowledge base as a set of sentences and formulas in first order logic, and introduces other basic concepts the 1st-order logic builds upon.
A *formula* consists of following elements: constants (the domain objects), variables (ranges over objects in the domain), functions (mappings from tuples of objects to objects) and predicates (relations between objects in the domain or objects attributes).
A *term* is defined as an expression representing an object in the domain.
Literals are defined like a predicate or its negation. A *clause* is defined as a disjunction of literals.
Grounding is known as the replacement of all variables by constants.
World (model, interpretation) is the assignment of truth values to all ground predicates
A formula is satisfiable iff there exists at least one world in which it is true.
The basic inference problem in first-order logic is to determine whether a knowledge base KB entails a formula F, i.e., if F is true in all worlds where KB is true (denoted by KB |= F).
Formulas are typically converted to a clausal form

(conjunctive normal form), which is a conjunction of clauses, being a disjunction of literals.
The inference in first-order logic is only semidecidable. Because of this, knowledge bases are often constructed using a restricted subset of first-order logic with more desirable properties. The Horn clauses, constructed containing the most positive literal are the most widely-spread approach.

C. Markov Logic

A logical KB is a set of hard constraints on the set of possible worlds: Let's make them soft constraints: When a world violates a formula, It becomes less probable, not impossible
Each formula is given a weight to indicate how strong the represented constraint is: The higher weight, the stronger the constraint

A Markov Logic Network (MLN) is a set of pairs (F, w) where

- F is a formula in first-order logic
- w is a real number

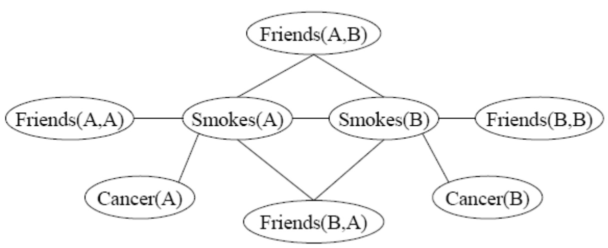
Together with a set of constants, it defines a Markov network with

- One node for each grounding of each predicate in the MLN
- One feature for each grounding of each formula F in the MLN, with the corresponding weight w

Coming back to the smoke and friendship sample: an MLN containing the formulas

$$\forall x \; Smokes(x) \Rightarrow Cancer(x)$$
$$\forall x, y \; Friends(x, y) \Rightarrow \big(Smokes(x) \Leftrightarrow Smokes(y)\big)$$

Smoking causes cancer and friends have similar smoking habits applied to the constants Anna and Bob (or A and B for short) yields the ground Markov network in Figure 1.
Notice that, although the two formulas above are false as universally quantified logical statements, as weighted features of an MLN they capture valid statistical regularities, and in fact represent a standard social network model.



D. Inference

A basic inference task is finding the most probable state of the world given some evidence. (This is known as MAP inference in the Markov network literature, and MPE inference in the Bayesian network literature.)

$$\max_y \; P(y \mid x)$$

This can be done using any weighted satisfiability solver, and (remarkably) need not be more expensive than standard logical inference by model checking

$$\max_y \; \frac{1}{Z_x} \exp\left(\sum_i w_i n_i(x, y) \right)$$

$$\max_y \; \sum_i w_i n_i(x, y)$$

This is just the weighted MaxSAT probable. Use weighted SAT solver (e.g., MaxWalkSAT, which is potentially faster than logical inference:

```
for i ← 1 to max-tries do
   solution = random truth assignment
   for j ← 1 to max-flips do
      if all clauses satisfied then
         return solution
      c ← random unsatisfied clause
      with probability p
         flip a random variable in c
      else
         flip variable in c that maximizes
            number of satisfied clauses
return failure
```

One problem with this approach is that it requires propositionalizing the domain (i.e., grounding all atoms and clauses in all possible ways), which consumes memory exponential in the arity of the clauses. The author's research group has overcome this by developing LazySAT, a lazy version of MaxWalkSAT which grounds atoms and clauses only as needed:

```
for i ← 1 to max tries do
  active atoms ← atoms in clauses not satisfied by DB
  active clauses ← clauses activated by active atoms
  soln ← a random truth assignment to active atoms
  cost ← sum of weights of unsatisfied clauses in soln
  for i ← 1 to max flips do
    if cost ≤ target then
       return "Success, solution is", soln
    end if
    c ← a randomly chosen unsatisfied clause
    if Uniform(0,1) < p then
       vf ← a randomly chosen variable from c
    else
       for each variable v in c do
         compute DeltaCost(v), using weighted KB if v doesn't
         belong to active atoms
       end for
       vf ← v with lowest DeltaCost(v)
    end if
   if vf doesn't belong to active atoms then
```

add vf to active atoms
add clauses activated by vf to active clauses
end if
soln ← soln with vf flipped
cost ← cost + DeltaCost(vf)
end for
end for
return “Failure, best assignment is”, best soln found

E. Learning

For the learning we need to starting assumptions: data is a relational database and the world is closed (otherwise EM)
There are two complementary approaches: learning parameters (weights) or learning structure (formulas)

1) Generative weight learning

This relational database consists of one or more “possible worlds” that form our training examples. Note that we can learn to generalize from even a single example because the clause weights are shared across their many respective groundings. We assume that the set of constants of each type is known. We also make a closed-world assumption: all ground atoms not in the database are false. This assumption can be removed by using an EM algorithm to learn from the resulting incomplete data. The gradient of the log-likelihood with respect to the weights is

$$\frac{\partial}{\partial w_i} \log P_w(x) = n_i(x) - E_w[n_i(x)]$$

where the sum is over all possible databases x′, and Pw(X = x′) is P(X =x′) computed using the current weight vector w = (w1, . . . ,*wi*, . . .). In other words, the *ith* component of the gradient is simply the difference between the number of true groundings of the *ith* formula in the data and its expectation according to the current model. The problem is the computational cost, because it requires inference at each step (very slow).

2) Discriminative Weight Learning

Maximize conditional likelihood of query (*y*) given evidence (*x*)

$$\frac{\partial}{\partial w_i} \log P_w(y \mid x) = n_i(x, y) - E_w[n_i(x, y)]$$

Approximate expected counts with:
counts in MAP state of *y* given *x* (with MaxWalkSAT)
with MC-SAT

3) Structure Learning

Generalizes feature induction in Markov nets.
Any inductive logic programming approach can be used, but the goal is to induce any clauses, not just Horn ones.
The evaluation function is intended to be the likelihood
Requires learning weights for each candidate, but it turns out not to be bottleneck, unlike the clause grounding counting.
To overcome the real bottleneck the counting of clause groundings presents, sub-sampling can be introduced

F. Application

The application spectrum of the Markov logic is very wide: information extraction (very successfully applied in biology), integration of probabilistic predictions. In the fields of web mining, activity recognition, natural language processing, computational biology, robot mapping and navigation, game playing and others are under way.

II. PLUSSES

The first strong point of this article is the schematic and very structured way Pedro introduced the two challenges of the AI, the existing approaches and how the Markov Logic combines the best of two disciplines to address the challenges.
Following this brilliant framing of the problematic, Pedro starts by explaining the fundamentals and also the limitations of both 1st-order logic and Markov networks, prior to deep diving into the Markov logic. This also speaks for the structured thinking of the author, as well as makes the reading and understanding of the article much easier.
The section regarding application areas was very adequate to show the practical applicability of the Markov Logic, rather than leaving it in a theoretical world.
Very positive was the last section, where Pedro provides an outlook about current and future research directions, intended somehow to spark the reader’s interest.

III. DELTAS

When the author gets into lower algorithms details, not all the required information is given. This is always a trade-off between making the article too long and repeating things or just providing references, but my point refers rather to the language used, sometimes too cryptic.
Another aspect I would rather keep out of this article is the exaggerated of the research group of the author with a complete section about the self-developed Alchemy system (from my point of view out of place).

IV. BIBLIOGRAPHIC REFERENCES CHECK

First of all a few words about the usage of citations: the author introduces the challenges, the different approaches up to the moment (each one given a reference to a core work), to get down later to the Markov Logic.

The 1st-order logic part, as well as the Markov network is provided with the necessary references (to works where the fundamentals are precisely explained).
When introducing the Markov Logic, the author quotes previous own works in this area (which allow for briefness, instead of overloading the text), but not leaving details unexplained.
The two more complex sections, which are inference and learning in MLN, are copiously provided with the references to sources where the algorithms the author is basing his finding on, are explained in detail.
When Pedro writes about the MLN applications, uses quite a few references to current/on-going project, which conveys an idea of the state-of-the-art of the industrial applicability of such learning models.
All references are appropriated at the publication time, and correctly placed in the article. Moreover, we could classify the references into following groups:

- References about fundamentals (provided when introducing concepts)
- References related to current research areas and state-of-the-art
- References intended to extend in further level of detail the explanation of a given topic.

V. Subsequent works related to MLN

Since the publication of this article (2006), there have been a lot of researches around MLN, mainly driven by Pedro Domingos' research group.
I will briefly summarize the core articles published subsequently to the article we are analyzing:

1) *Entity resolution with Markov Logic [1]*

This paper proposes a unifying framework for entity resolution. The team shows how a small number of axioms in Markov logic capture the essential features of many different approaches to this problem, in particular non-i.i.d. ones, as well as the original Fellegi-Sunter model. Experiments on two citation databases evaluate the contributions of these approaches, and illustrate how Markov logic enables us to easily build a sophisticated entity resolution system.

2) *Recursive random fields and MLN [2]:*

Recursive random fields overcome some salient limitations of Markov logic. While MLNs only model uncertainty over conjunctions and universal quantifiers, RRFs also model uncertainty over disjunctions and existentials, and thus achieve a deeper integration of logic and probability. Inference in RRFs can be carried out using Gibbs sampling and iterated conditional modes, and weights can be learned using a variant of the back-propagation algorithm.
The main disadvantage of RRFs relative to MLNs is reduced understandability. One possibility is to extract MLNs from RRFs with techniques similar to those used to extract propositional theories from KBANN models. Another important problem for future work is scalability. Here we plan to adapt many of the MLN optimizations to RRFs. Most importantly, we intend to apply RRFs to real datasets to better understand how they work in practice, and to see if their greater representational power yields better models.

3) *Statistical Predicate Invention [3]*

The research group proposed statistical predicate invention, the discovery of new concepts, properties and relations in structured data, as a key problem for statistical relational learning. They then introduced MRC, an approach to SPI based on second-order Markov logic.
MRC forms multiple relational clusterings of the symbols in the data and iteratively refines them. Empirical comparisons with a Markov logic structure learning system and a state-of-the-art relational clustering system on four datasets show the promise of their model.
They speculate that all relational structure learning can be accomplished with SPI alone. Traditional relational structure learning approaches like ILP build formulas by incrementally adding predicates that share variables with existing predicates. The dependencies these formulas represent can also be captured by inventing new predicates. For example, consider a formula that states that if two people are friends, either both smoke or neither does
SPI can compactly represent this using two clusters, one containing friends who smoke, and one containing friends who do not. The model we introduced in this paper represents a first step in this

4) *Markov logic in infinite domains [4]*

In this paper, the Domingos and Singla extended the semantics of Markov logic to infinite domains using the theory of Gibbs measures. They gave sufficient conditions for the existence and uniqueness of a measure consistent with the local potentials defined by an MLN. They also described the structure of the set of consistent measures when it is not a singleton, and showed how the problem of satisfiability can be cast in terms of MLN measures. Directions for future work include designing lifted inference and learning algorithms for infinite MLNs, deriving alternative conditions for existence and uniqueness, analyzing the structure of consistent measure sets in more detail, extending the theory to non-Herbrand interpretations and recursive random fields, and studying interesting special cases of infinite MLNs.

5) *Efficient Weight Learning for MLN [5]*

Weight learning for Markov logic networks can be extremely ill-conditioned, making simple gradient descent-style algorithms very slow to converge. In this paper we studied a number of more sophisticated alternatives, of which the best-performing one is preconditioned scaled conjugate gradient. This can be attributed to its effective use of second-order information. However, the simple heuristic of dividing the learning rate by the true clause counts for each weight can sometimes give very good results. Using one of these methods instead of gradient descent can yield a much better model in less time.

References

[1] Domingos, P. and Singla, P. Entity Resolution with Markov Logic,. Proceedings of the Sixth IEEE International Conference on Data Mining (pp. 572-582), 2006. Hong Kong: IEEE Computer Society Press.

[2] Domingos, P. and Lowd D. Recursive Random Fields. Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (pp. 950-955), 2007. Hyderabad, India: AAAI Press.

[3] Domingos, P. and Kok, S. Statistical Predicate Invention, Proceedings of the Twenty-Fourth International Conference on Machine Learning (pp. 433-440), 2007. Corvallis, Oregon: ACM Press

[4] Domingos, P.and Singla, P. Markov Logic in Infinite Domains. Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (pp. 368-375), 2007. Vancouver, Canada: AUAI Press

[5] Domingos, P. and Lowd, D. Efficient Weight Learning for Markov Logic Networks. Proceedings of the Eleventh European Conference on Principles and Practice of Knowledge Discovery in Databases (pp. 200-211), 2007. Warsaw, Poland: Springer.