Applying Markov Logic to Document Annotation and Citation Deduplication

Jean Baptiste Faddoul*, Boris Chidlovskii*

*Xerox Research Centre Europe
6, chemin de Maupertuis 38240 Meylan-FRANCE

Abstract. Structured learning approaches are able to take into account the relational structure of data, thus promising an enhancement over non-relational approaches. In this paper we explore two document-related tasks in relational domains setting, the annotation of semi-structured documents and the citation deduplication. For both tasks, we report results of comparing relational learning approach namely *Markov logic*, to non-relational one namely *Support Vector Machines (SVM)*. We discover that increased complexity due to the relational setting is difficult to manage in large scale cases, where non-relational models might perform better. Moreover, our experiments show that in Markov logic, the contribution of its probabilistic component decreases in large scale domains, and it tends to act like First-order logic *(FOL)*.

1 Introduction

A large majority of existing machine learning models can be seen as non-relational. They represent each object as an isolated point in a space, and they learn prediction models using the features of each object. A new trend in statistical machine learning is represented by relational models that take into account the relational structure of data. The relations between objects are frequent in real world cases, and taking them into account may offer a potential performance improvement over non-relational models. On the other hand, real world data sets are large in the number of objects, the feature dimensions and even the relation numbers. So, from the scalability point of view, simpler non-relational models might scale better and might outperform complex ones. In this paper, we study relational and non-relational models on two different tasks, the annotation of semi-structured documents and the citation deduplication.

In our experiments, we use SVM as the best state of art non-relational model. As relational models, we applied *Markov logic* (introduced in Domingos and Richardson (2007)), which is a *Statistical Relational Learning (SRL)* approach that combines FOL and Markov networks. Markov logic has been chosen for its capacity to represent more complex relations than previously used models and a number of important results shown when tested on different domains (Culotta and McCallum (2006) and Kok and Yih (2009)).

1.1 Annotation of Semi-Structured Documents

Document annotation can be seen as a *collective classification*[1] task (we refer the details to Chakrabarti et al. (1998) and Neville and Jensen (2003)). Objects to be classified are fragments of a document such as tokens, lines and paragraphs, while classes are semantic labels of these fragments. Each object is described by a set of features and a class represents its semantic label (title, author, reference, etc.). Additionally, various relations between two arbitrary objects can be identified, like *next_token(), same_paragraph()*, etc.

1.2 Citation Deduplication

Citation deduplication is the problem of determining records in a database referring to the same real-world entity (Monge and Elkan (1996)). Each record in the database is composed of multiple fields. Information inferred from the record matching can be propagated to field matching and vice versa.

We introduced the difference between relational and non-relational models and described two tasks in a relational domains setting. In the next section we present Markov logic. In Section 3 we report results of the comparison between Markov logic and SVM models on both tasks. Section 4 discusses the scalability issues and Section 5 concludes this paper.

2 Markov Logic

Markov logic is a Structured Relational Learning model that combines FOL and probabilistic graphical models. It has been introduced by Domingos and Richardson (2007).

In FOL, a KB is a set of formulas that can be seen as a set of constraints on the set of possible worlds. If a possible world violates one formula, it has zero probability. The basic idea in Markov logic is to soften these constraints: when a possible world violates one formula in the KB it is less probable, but not impossible. The fewer formulas a world violates, the more probable it is. Each formula has an associated weight that reflects how strong a constraint it is: the higher the weight, the greater the difference in log probability between a world that satisfies the formula and one that does not, other things being equal. A set of formulas in Markov logic is called a *Markov Logic Network (MLN)*. $MLNs$ define probability distribution over possible worlds [2], where each state of the Markov network $M_{L,C}$ represents a possible world.

3 Experiment Settings and Results

3.1 Annotation Task

For the annotation task we used the *BizCard* collection, which is a collection of scanned business cards with different layouts. Each card is segmented into blocks and lines, where each line is segmented into tokens (each one has a semantic annotation) and separators that do not have annotation $(-, ., , etc.)$. Each token is annotated with one of 17 classes, such as address, name, email, affiliation, etc. The collection contains 106 business cards with an average of

[1] The objects' classes are not independents given the observations (the features).

[2] A possible world is a truth assignment for all groundings of all predicates in the *Knowledge Base (KB)*.

30 tokens to be annotated in each card. Each token is described with 135 features which have been defined by an expert, the features are classified in three groups:

Token content features (e.g. number of digits in a token), Token attribute features (e.g. e.g., the font type) and Line content features (e.g. containment of a country name in a certain line)

3.1.1 Models

We compared a non-relational approach namely SVM with a relational one namely Markov logic:

SVM: we deploy the LibSVM package by Chang and Lin (2001). We learn SVM models with the linear kernel, because training with different types of kernels reports that the linear one shows the best performance in high dimensional domains.

As features we used all 135 features extracted from the dataset. At the pre-processing step, we rescale finite float value features into the [0,1] range. Additionally, categorical features (like color) have been mapped into a set of boolean features (isBlue, isGreen, etc.).

Markov logic: For training Markov logic models we deploy the Alchemy software [3]. As the Markov logic adapts the syntax of FOL, representing the annotation task requires a KB definition. The predicates of such a KB have the following patterns:

- $Feat(x, v)$**:** the token x has value v for the feature $Feat$. The domain of x is the set of tokens in the training set during training, and in the testing set during testing. The domain of v is the set of possible values for the feature $Feat$. The feature $Feat$ may be a local feature in the token x itself, or a feature in another token x' being in a certain relation with x. There are 135 feature predicates of this type.

- $Class(x, c)$**:** the token x has class c. c's domain is the set of all possible annotations.

- $Rel(x, x')$**:** a relation Rel exists between x and x'. For our experiments, The predicate Rel could be one out of six predicates, each of which is defined on a pair of tokens: Left Brother $LB(x_1, x_2)$, Second Left Brother $2LB(x_1, x_2)$, Right Brother $RB(x_1, x_2)$, Second Right Brother $2RB(x_1, x_2)$, Next Line $NL(x_1, x_2)$ and Previous Line $PL(x_1, x_2)$.

We run several tests of MLNs for this task. In each run, we used different sets of features and relations. Two formula templates were used:

1. Non-Relational formulas: $Feat(x, v) \Rightarrow Class(x, c)$, that models the classification problem depending only on features only (without relations).

2. Relational formulas: $Class(x_1, c_1) \wedge Rel(x_1, x_2) \Rightarrow Class(x_2, c_2)$, that models the classes based on both relations between tokens and their classes.

3.1.2 Results

Table 1 reports results of MLN experiments on the BizCard collection. In all runs, we use the 5-fold cross validation. In each run a different set of relations have been used along with

[3] http://alchemy.cs.washington.edu/

all the 135 features. A large number of runs have been done, those with the highest *accuracy*[4] are reported in the Table 1. The baseline case with no relations used gives 66.42% of accuracy. Relational formulas improve the accuracy when using the small depth ones, with higher depth (2-RB, 2-LB) we observe the accuracy loss.

Relations	Accuracy
none	66.42
LB	**72.85**
RB	70.78
LB + RB	67.82
LB + 2-LB	66.87
NL+PL	66.56

TAB. 1 – *Accuracy results on BizCard using different settings of Markov logic.*

Table 2 compares accuracy values and the running time in the 5-fold cross validation mode for both SVM and the MLN that has the best results. As one can see SVM outperforms MLN in both accuracy and running time.

3.2 Citation Deduplication Task

3.2.1 CORA Data Set

CORA is a collection of 1879 different citations in Computer Science research papers [5]. It contains citations to 168 different research paper, so there is an average of 11 citations to the same paper. Each citation is segmented into fields (author, title, publisher, year, etc.).

3.2.2 Models

As in the case of Business card, we compared an SVM classifier with an MLN classifier:

SVM: we used LibSVM by Chang and Lin (2001) to implement a binary classifier based on Levenshtein distance [6] as follow. For each pair of citations we calculate the Levenshtein distance between each field type (author, year, title, etc.). We obtain then a distance vector for

[4]The percentage of correctly classified examples

[5]Available at http://www.cs.umass.edu/∼mccallum/data/cora-refs.tar.gz

[6]Minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character

	Accuracy	Running Time
SVM 135 Features	78.42	0.3 hours
MLN 135 Feat. + LB relation	72.85	5.1 hours

TAB. 2 – *SVM and MLN Accuracy and Running time BizCard.*

each pair of citations. This distance vector is a vector in the SVM space with class 1 if the two citations are the same, and class 0 if not. So the task is turned into a binary classification task.

Markov logic: We cite results by Singla and Domingos (2006). The KB used contains three types of predicates (we refer the detailed description of the KB to their paper):

1. Class predicates are the predicates that should be predicted:
   $Author(b, a)$, $Title(b, t)$, $Venue(b, v)$.
2. Evidence predicates are the observed predicates:
   $HasWordAuthor(a, w)$, $HasWordTitle(t, w)$, $HasWordVenue(v, w)$.
3. Match predicates are equality predicates between fields (they must be associated with the axioms of equality):
   $SameAuthor(a_1, a_2)$, $SameBib(b_1, b_2)$, $SameTitle(t_1, t_2)$, $SameVenue(v_1, v_2)$.

The formulas in the KB model the relations by connecting evidence and class predicates with match predicates.

	AUC	Running Time
SVM	97.88	0.2 hours
MLN	98.01	4.2 hours

TAB. 3 – *AUC and Running time on CORA.*

3.2.3 Results

Table 3 compares MLN and SVM. We used *AUC* (area under the precision-recall curve) to be able to compare our results with Domingos and Singla's best results. Running time showed in the table is obtained by running a 5-fold cross validation on the data. We see that the AUC for MLN and SVM is comparable, but with a significant difference in running time.

4 Discussion

As our evaluations show, MLNs do not scale as well as simpler models like SVMs. This is likely to happen when using complicated relations or very large datasets, which yields in a very large generated Markov Networks. Actually, FOL is an MLN with infinite weight values, because in FOL every formula is an infinitely hard constraint, so a world cannot violate any formula (proven in Domingos and Richardson (2007)). In our experiments the tendency of MLN to have very large weight values and so acting like FOL was obvious when we used complicated models with large scale data sets. Table 4 compares the difference in average weight values for the BizCard collection when we change the size of the domain of constants.

The role of logic in MLNs is just at the representation level, in order to obtain small granularity in knowledge representation. Whereas, at the inference level, the logical inference is appealing but it is replaced by probabilistic inference. As experiments shows, this representation ability provided by FOL, complicates the graph making the scalability a harder task.

Fraction of corpus used	Average weights
10%	1.03
50%	50.3
100%	807.01

TAB. 4 – *Average weights on different subsets of BizCard.*

5 Conclusion

Markov logic is a structure learning model able to model complex relations between objects to better catch the complexity of real world data. Our experiments confirm that modeling object relations and using them in the learning can improve performance on document-relevant tasks. Nevertheless, its performance remains modest with respect to the best non-relational models. Another drawback is its lake of scalability. Such a drawback is caused by the very large size of generated Markov networks. A possible solution to the problem is to combine relational and non-relational models, in such a way that non-relational models trained with object features will feed relational models trained with such predictions and relations only.

References

Chakrabarti, S., B. Dom, and P. Indyk (1998). Enhanced hypertext categorization using hyperlinks.

Chang, C.-C. and C.-J. Lin (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Culotta, A. and A. McCallum (2006). Practical markov logic containing first-order quantifiers with application to identity uncertainty. In *CHSLP '06: Proceedings of the Workshop on Computationally Hard Problems and Joint Inference in Speech and Language Processing*, Morristown, NJ, USA, pp. 41–48. Association for Computational Linguistics.

Domingos, P. and M. Richardson (2007). Markov logic: A Unifying Framework for Statistical Relational Learning. In L. Getoor and B. Taskar (Eds.), *Introduction to Statistical Relational Learning*, pp. 339–371. MIT Press.

Kok, S. and W.-T. Yih (2009). Extracting product information from email receipts using markov logic. In *Proceedings of the Sixth Conference on Email and Anti-Spam*.

Monge, A. and C. Elkan (1996). The field matching problem: Algorithms and applications. In *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 267–270.

Neville, J. and D. Jensen (2003). Collective classification with relational dependency networks. *Journal of Machine Learning Research 8*, 2007.

Singla, P. and P. Domingos (2006). Entity resolution with markov logic. In *In ICDM*, pp. 572–582. IEEE Computer Society Press.