Towards Learning Optimal Markov Logic Networks for Sequence Labeling

Naveen Nair[123], Ajay Nagesh[123], and Ganesh Ramakrishnan[21]

[1] IITB-Monash Research Academy, Old CSE Building, IIT Bombay
[2] Department of Computer Science and Engineering, IIT Bombay
[3] Faculty of Information Technology, Monash University
{naveennair,ajaynagesh,ganesh}@cse.iitb.ac.in

Abstract. Discovering relational structure between input features in sequence labeling models has been shown to improve their accuracies in several problem settings [1–3]. In [4], we proposed an approach using Hierarchical Kernels that learns optimal feature conjunctions for the struct-svm objective, which is referred to as StructRELHKL. Although this approach yields optimal models in propositional space, its extension to complex first order settings is non-trivial and challenging. In this paper, we look into leveraging that work to discover first-order features for sequence labeling. We formalize our problem as that of simultaneously learning the structure and parameters of a Markov Logic Network (MLN) for sequence labeling, which we abbreviate as Markov Logic Chains (MLCs). We categorize first order MLN features based on their complexity and show that complex features can be constructed from simpler ones. We define a self-contained class of features called candidate definition ($\mathcal{CD}$) features which can be conjoined to yield MLN features. Our approach first generates a set of relevant $\mathcal{CD}$s and then makes use of the algorithm for StructRELHKL to learn conjunctions of $\mathcal{CD}$s that are globally optimal for the StructRELHKL objective. On this account, our work is a progression towards optimally learning the structure as well as parameters of MLNs for sequence labeling tasks (as against learning them separately and/or greedily).

1 Introduction

Sequence labeling is the task of assigning a class label to each instance in a sequence of observations. Typical sequence labeling algorithms learn probabilistic information about the neighboring states along with the probabilistic information about the observations. Recent works, including ours, have looked into the problem of learning better sequence labeling models by discovering the relational structure between input features [1–3]. However since these approaches employ a greedy search to discover useful input features, an optimal model is not guaranteed. In [4], we proposed Rule Ensemble Learning using Hierarchical Kernels in Structured Output Spaces (StructRELHKL) for learning optimal models for sequence labeling. In such settings, training objective is to learn features that make the score, $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, of original output sequence $\mathcal{Y}$

greater than any other possible output sequence, given an input sequence $\mathcal{X}$. The score is defined as, $F(X, Y; \mathbf{f}) = \langle \mathbf{f}, \boldsymbol{\psi}(X, Y) \rangle$, where $\boldsymbol{\psi}$ is the feature vector (features describing observation structure and transitions), and $\mathbf{f}$ is the weight vector. Inference is performed by the decision function $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ defined by $\mathcal{F}(X; \mathbf{f}) = \arg\max_{Y \in \mathcal{Y}} F(X, Y; \mathbf{f})$. StructRELHKL exploits the hierarchical structure in the (exponential) feature space and efficiently learns a sparse set of simple propositional features and their weights. Although this approach yields optimal models in propositional space, due to the inherent complexity in first order settings such as huge number of groundings, variable sharing, background knowledge, subsume equivalence *etc.*, the first order extension is non-trivial and is a challenging problem.

In this paper, we look into leveraging StructRELHKL to discover first order features. To this end, we present sequence labeling problem in a first order Markov Logic Network (MLN) [5] framework. We categorize first order MLN features based on their complexity and identify the class of features that can be efficiently constructed from simpler ones by StructRELHKL[4]. We identify a self-contained class of features called candidate definition ($\mathcal{CD}$) features as building blocks, whose unary/multiple conjunctions result in the final model. Our approach first generates $\mathcal{CD}$s that cover a threshold number of examples (weak relevance) and employs the StructRELHKL algorithm to simultaneously learn optimal conjunctions of $\mathcal{CD}$s and their weights (as against learning them separately and/or greedily). We learn optimal MLNs with respect to weakly relevant $\mathcal{CD}$s and therefore, our work is a progression towards optimal learning of MLNs for sequence labeling tasks. We refer to our setting as Markov Logic Chains (MLC). We give a brief introduction to general MLN settings below before concluding the section.

MLNs are typically represented as a collection of first order clauses with real valued weights attached to each clause. Conventional MLN systems tend to learn the structure and parameters separately. There have been a few approaches recently to learn the structure and parameters simultaneously [6–8]. However since the feature space is exponential (possibly infinite), all the MLN structure learning approaches are greedy and thus cannot guarantee optimal models. Learning optimal MLNs is a hard task in general settings and is harder in sequence labeling tasks. In this paper, we propose an approach that optimizes a substancial part of MLN structure learning. The paper is organized as follows.

In section 2, the complexity based categorization of features is discussed. We discuss our approach in section 3 and conclude the paper in section 4.

2 First Order Definite Features for Markov Logic Chains

We start the section by defining the categories of predicates and then discuss the complexity based classification of features.

[4] In this work, we restrict our discussion to function-free first order definite features. Moreover, since a class specific feature can be constructed by conjoining the body literals of a definite clause whose head depicts the class label, we use the terms first order definite clause and first order feature interchangeably

Similar to *structural* and *property* predicates in 1BC clauses [9], we define two types of predicates, *viz. (inter) relational* and *quality* predicates. A *relational* predicate is a binary predicate that represents the relationship between *types* or between a *type* and its parts, where a *type* is an entity described by its attributes. A *quality* predicate is a predicate that reveals a property of a part (or subset) of a *type*. From the example clauses given below, `contains(X,X1)`, `before(X1,X2)` are *relational* predicates and all other predicates are *quality* predicates.

1. `nerLOC(X) :- contains(X,X1), countrydict(X1)`
2. `nerLOC(X) :- contains(X,X1)`
3. `nerPER(X) :- contains(X,X1), personfullname(X1), caps(X1)`
4. `nerPER(X) :- contains(X,X1), fname(X1), contains(X,X2), before(X1,X2), lname(X2), caps(X2)`
5. `nerLOC(X) :- contains(X,X1), countrydict(X1), contains(X,X2), regiondict(X2)`

We now classify first order definite features below.

Candidate Definitions ($\mathcal{CD}$): In $\mathcal{CD}$ features, new local variables can only be introduced in a *relational* predicate, where a local variable is a variable not present in the head predicate. Unlike in 1BC clauses, any number of new local variables can be introduced in a *relational* predicate. Any number of *relational* and *quality* predicates can be conjoined to form a $\mathcal{CD}$ such that the resultant $\mathcal{CD}$ is minimal and the local variables introduced in *relational* predicates are consumed by some other *relational* or *quality* predicates. Here a minimal clause is one which cannot be constructed from smaller clauses that share no common variables other than that in the head. So clauses 1, 3 and 4 above are $\mathcal{CD}$s whereas clauses 5 (not minimal) and 2 (variable `X1` is not consumed) are not.

Primary Features ($\mathcal{PF}$): $\mathcal{PF}$s are $\mathcal{CD}$ clauses that have at-most one *quality* predicate for every new local variable introduced. This is similar to elementary features in [9] except that elementary features allow only one new local variable in a *structural* predicate. Only clause 1 above is a $\mathcal{PF}$.

Candidate Refinements ($\mathcal{CR}$): A $\mathcal{CR}$ is a type of definite clause formed by the conjunction of one or more $\mathcal{CD}$s without unification of body literals. Only the head predicates are unified. As in $\mathcal{CD}$s, every local variable introduced in a $\mathcal{CR}$ should be consumed. Clauses 1 , 3 , 4 and 5 are $\mathcal{CR}$s whereas 2 is not.

First Order Definite Features ($\mathcal{DF}$): First order definite features are features with none of the above restrictions. Therefore, all the given examples are $\mathcal{DF}$s.

We now state some of the relationships between these types of clauses. Some of the proofs are skipped due to the space restrictions.

Lemma 1. The set of primary features is a proper subset of the set of Candidate Definitions. That is, $\mathcal{PF} \subset \mathcal{CD}$.

Lemma 2. The set of candidate definitions is a proper subset of the set of candidate refinements. That is, $\mathcal{CD} \subset \mathcal{CR}$.

Lemma 3. The set of candidate refinements is a proper subset of the set of full first order definite features. $\mathcal{CR} \subset \mathcal{DF}$.

Lemma 4. Every $\mathcal{CD}$ can be constructed from $\mathcal{PF}$s using unifications.

Proof. The difference a $\mathcal{CD}$ has with $\mathcal{PF}$ is that it can have more than one *quality* predicates for each local variable introduced. Let l_p be a *relational* literal in the body of a $\mathcal{CD}$ clause which introduces only one local variable. Let

$l_1, l_2, \ldots, l_{p-1}$ be the set of *relational* literals in the body, which l_p depends on. Let there are $n \geq 0$ number of dependency chains starting from l_p to a *quality* predicate, each of which is represented as $l^i_{p+1}, \ldots, l^i_k$. We define l_p as a pivot literal if $n > 1$. For simplicity, we assume there is only one pivot in a clause. Now, we can construct n $\mathcal{PF}$ clauses from this with the body of the i^{th} clause as $l_1, l_2, \ldots, l_{p-1}, l_p, l^i_{p+1}, \ldots, l^i_k$, where l^i_k is a *quality* predicate. It is trivial to see that these n clauses can be unified to construct the original $\mathcal{CD}$. For multiple pivot literal clauses, the above method can be applied recursively until $\mathcal{PF}$ clauses are generated. The proof can be extended to pivot literals with multiple new local variables by using a dependency tree structure in place of chain.

Lemma 5. Every $\mathcal{CR}$ can be constructed from $\mathcal{CD}$s by conjunctions.

Lemma 6. $\mathcal{CR}$s are first order $\mathcal{DF}$s with local variable reuse restriction.

3 Optimal Structure and Parameter Learning for MLC

We start this section with a brief discussion on Markov Logic Chains.

Markov Logic Chains: capture state-observation relations and state transition relations in the form of weighted first order clauses. Similar to MLNs [5], MLCs define a probability distribution over a possible world I as, $P(I|H,B) = \frac{1}{Z} \prod_{C \in H \cup B} \phi_C(I)^{n_C(I)}$, where H is the hypothesis, B is the background knowledge, $\phi_C(I) = e^{f_C}$, f_C is the weight attributed to the clause C, $n_C(I)$ is the number of true groundings of C in I and Z is the normalization constant. Therefore, an ideal MLC should have hypothesis $H^* = \arg\max_H P(I|H,B)$ for true interpretations I. We now briefly introduce our optimal feature induction approach in propositional settings before going into details of learning a first order MLC.

StructRELHKL: Our recent work, Rule Ensemble Learning using Hierarchical Kernels in Structured Output Spaces [4], utilizes the hierarchical structure of the exponential observation space and learns optimal and sparse features for sequence labeling tasks (please refer [4] for a complete description). However this approach is for propositional settings and cannot be trivially extended to first order settings. Moreover, it relies on the summability of kernels over descendants in polynomial time. In contrast, MLC domain has an ordering of first order features. Specializations and generalizations in first order space are done by unifications and anti-unifications respectively. This makes it hard to sum the descendant kernels in polynomial time. The complexities involved with background knowledge, unification, anti-unification, subsume equivalence *etc.* makes first order feature learning harder. We now look into the class of features that can be constructed using StructRELHKL from simpler ones.

MLC feature: As defined in section 2, $\mathcal{CR}$s can be constructed from unary/multiple conjunctions of $\mathcal{CD}$s without unifications. Hence we can apply StructRELHKL on $\mathcal{CD}$s to find their optimal conjunctions ($\mathcal{CR}$). Therefore, the space of $\mathcal{CR}$s can be defined as a partial order over conjunctions of $\mathcal{CD}$s[5]. With proper lan-

[5] Although the space of $\mathcal{CR}$s is a partial order over conjunctions and unifications of $\mathcal{PF}$s, since the kernels over descendants are not summable in polynomial time, StructRELHKL cannot be applied in this ordering. Whereas it is summable in the conjunction ordering over $\mathcal{CD}$s

guage restrictions, $\mathcal{C}D$s can be generated by ILP methods. StructRELHKL can be employed to find optimal conjunctions of $\mathcal{C}D$s. The next paragraph gives some insight into the optimality of the resultant MLC.

We define a $\mathcal{C}D$ as strongly relevant if it is constructed from optimal unifications of $\mathcal{P}F$s, which is a hard task. On the other hand, we consider a feature to be weakly relevant if it covers at-least a threshold percentage of examples. We are interested in $\mathcal{C}D$s that are at-least weakly relevant and our approach learns optimal MLCs with respect to weakly relevant $\mathcal{C}D$s. We now derive our approach in the following paragraph.

Towards learning optimal MLCs: The objective is to first learn self contained weakly relevant $\mathcal{C}D$s by employing ILP methods and then learn optimal conjunctions of $\mathcal{C}D$s and their weights simultaneously by leveraging the StructRELHKL approach. Since there is a plethora of literature available on ILP approaches for searching clauses with language restrictions [10, 11], we skip the discussion on generating $\mathcal{C}D$s and move on to give an overview of StructRELHKL [4] framework to construct $\mathcal{C}R$s.

An MLC should have features defining transition dependencies between the states as well as the observation dependencies. Let $\boldsymbol{\psi_T}$ represent the feature vector corresponding to all transition features and $\boldsymbol{\psi_{CR}}$ represent the feature vector corresponding to all observation features (space of all possible conjunctions of $\mathcal{C}D$s). For the sake of visualization, assume there is a lattice for each label in a multi-class setup. As defined in introduction section, $\boldsymbol{\psi}$ represents a combination of $\boldsymbol{\psi_T}$ and $\boldsymbol{\psi_{CR}}$. We assume that both $\boldsymbol{\psi_{CR}}$ and $\boldsymbol{\psi_T}$ are of dimension equal to the dimension of $\boldsymbol{\psi}$ with zero values for all elements not in their context. In similar spirit, the feature weight vector $\mathbf{f}$ can be constructed from $\mathbf{f_{CR}}$ and $\mathbf{f_T}$. Similarly, $\mathcal{V}$, the indices of the elements of $\boldsymbol{\psi}$, can be constructed from $\mathcal{V}_{\mathbf{CR}}$ and $\mathcal{V}_{\mathbf{T}}$. Our objective is to simultaneously select a sparse set of $\mathcal{C}R$s and their weights along with the transition feature weights, which is presented below.

$$\begin{aligned}
&\min_{\mathbf{f},\boldsymbol{\xi}} \frac{1}{2}\Omega_{CR}(\mathbf{f_{CR}})^2 \;+\; \frac{1}{2}\Omega_{T}(\mathbf{f_T})^2 \;+\; \frac{C}{m}\sum_{i=1}^{m}\xi_i, \\
&\forall i, \forall Y \in \mathcal{Y} \setminus Y_i : \; \langle \mathbf{f}, \boldsymbol{\psi}_i^{\delta}(Y)\rangle \geq 1 - \frac{\xi_i}{\Delta(Y_i, Y)} \\
&\forall i : \; \xi_i \geq 0
\end{aligned} \tag{1}$$

where $\Omega_{CR}(\mathbf{f_{CR}})$ is defined in [12] as $\sum_{v \in \mathcal{V}_{\mathbf{CR}}} d_v \parallel \mathbf{f_{CR}}_{D(v)} \parallel_{\rho}$, $\rho \in (1, 2]$, $d_v \geq 0$ is a prior parameter showing usefulness of the feature conjunctions, $D(v)$ represents the set of descendants (including itself) of node v in the partial order, $\mathbf{f_{CR}}_{D(v)}$ is the vector with elements as $\parallel f_{CRw} \parallel_2 \;\; \forall w \in D(v)$, and $\parallel . \parallel_{\rho}$ represents the ρ-norm , $\Omega_T(\mathbf{f_T})$ is the 2-norm regularizer $\big(\sum_i f_{Ti}^2\big)^{\frac{1}{2}}$, m is the number of examples, C is the regularization parameter, ξ's are the slack variables introduced to allow errors in the training set in a soft margin SVM formulation, $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$ represent the i^{th} input and output sequence respectively, $\langle \mathbf{f}, \boldsymbol{\psi}_i^{\delta}(Y)\rangle$ represents the value $\langle \mathbf{f}, \boldsymbol{\psi}(X_i, Y_i)\rangle - \langle \mathbf{f}, \boldsymbol{\psi}(X_i, Y)\rangle$, and $\Delta(Y, \hat{Y})$ represents the loss when true output is Y and the prediction is $\hat{Y}$.

The 1-norm in $\Omega_{CR}(\mathbf{f_{CR}})$ forces many of the $\| \mathbf{f_{CR}}_{D(v)} \|_\rho$ to be zero. Even in cases where $\| \mathbf{f_{CR}}_{D(v)} \|_\rho$ is not forced to zero, the ρ-norm forces many of node v's descendants to zero. This ensures a sparse and simple set of features. The problem is solved by an active set algorithm that incrementally adds features to active set until a sufficiency condition for optimality is satisfied.

4 Conclusion

Discovering the input structure in sequence labeling problems has shown to improve the performance in terms of accuracy. The recently introduced StructRELHKL approach learns optimal and sparse set of features in propositional settings, but has limitations in discovering first-order features. In this paper, we categorized first order features based on complexity and identified the class of features that can be constructed using StructRELHKL from simpler ones. Our approach generates candidate definition features (CD) and employs StructRELHKL to learn optimal conjunctions of CDs.

References

1. McCallum, A., Li, W.: Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In: HLT-NAACL. (2003)
2. Nair, N., Ramakrishnan, G., Krishnaswamy, S.: Enhancing activity recognition in smart homes using feature induction. International Conference on Data Warehousing and Knowledge Discovery (2011)
3. Mauro, N.D., Basile, T.M.A., Ferilli, S., Esposito, F.: Feature construction for relational sequence learning (2010)
4. Nair, N., Saha, A., Ramakrishnan, G., Krishnaswamy, S.: Rule ensemble learning using hierarchical kernels in structured output spaces. AAAI 2012 (2012)
5. Richardson, M., Domingos, P.: Markov logic networks. Mach. Learn. **62**(1-2) (February 2006) 107–136
6. Kok, S., Domingos, P.: Learning the structure of markov logic networks. In: Proceedings of the 22nd international conference on Machine learning. ICML '05, New York, NY, USA, ACM (2005) 441–448
7. Biba, M., Ferilli, S., Esposito, F.: Structure learning of markov logic networks through iterated local search. In: Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence, Amsterdam, The Netherlands, The Netherlands, IOS Press (2008) 361–365
8. Khot, T., Natarajan, S., Kersting, K., Shavlik, J.: Learning markov logic networks via functional gradient boosting. In: Proceedings of the 2011 IEEE 11th International Conference on Data Mining. ICDM '11, Washington, DC, USA, IEEE Computer Society (2011) 320–329
9. Flach, P.A., Lachiche, N.: First-order bayesian classification with 1bc (2000)
10. Nienhuys-Cheng, S.H., Wolf, R.d.: Foundations of Inductive Logic Programming. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1997)
11. Getoor, L., Taskar, B.: Statistical relational learning. MIT Press (2006)
12. Jawanpuria, P., Jagarlapudi, S.N., Ramakrishnan, G.: Efficient rule ensemble learning using hierarchical kernels. ICML (2011)