Distributed Relational State Representations for Complex Stochastic Processes (Extended Abstract)

Ingo Thon INGO.THON@CS.KULEUVEN.BE
Department of Computer Science, Katolieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Heverlee, Belgium

Kristian Kersting KERSTING@CSAIL.MIT.EDU
CSAIL, Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA, 02139-4307, USA

Abstract

Several promising variants of hidden Markov models (HMMs) have recently been developed to efficiently deal with large state and observation spaces and relational structure. Many application domains, however, have an apriori componential structure such as parts in musical scores. In this case, exact inference within relational HMMs still grows exponentially in the number of components. In this paper, we propose to approximate the complex joint relational HMM with a simpler, distributed one: k relational hidden chains over n states, one for each component. Then, we iteratively perform inference for each chain given fixed values for the other chains until convergence. Due to this structured mean field approximation, the effective size of the hidden state space collapses from n^k to n.

1. Introduction

In recent years, Statistical Relational Learning (SRL) has emerged as an active research subfield of Machine Learning. It is a relatively young research field that deals with machine learning and data mining in relational domains where observations may be missing, partially observed, and/or noisy. So far, however, surprisingly few SRL approaches have been developed for modeling dynamic domains, i.e., domains with temporal and/or sequential aspects. One reason might be that time is not simply yet another relation. The algorithmic complexity for general purpose, dynamic SRL approaches easily explodes and becomes intractable in practice if not quite strong assumptions are made such as low branching factors to keep tractability (Sanghai et al., 2003). Another alternative way to keep dynamic SRL approaches tractable is to lift simple dynamic probabilistic model, which naturally restrict the dynamics of the domain. This approach has been followed by Anderson et al. (2002) and by Kersting et al. (2006), who lifted (hidden) Markov models to the relational case. Hidden Markov models (HMMs) (Rabiner, 1989) are extremely popular for modeling dynamic domains. Application areas include computational biology, user modelling, speech recognition, empirical natural language processing, and robotics.

Many application domains, however, have an apriori componential structure such as parts in musical scores. In this case, exact inference within relational HMMs still grows exponentially in the number of components due to the combinatorical nature of the state space. In the propositional case, this 'curse of compositionality' has been successfully addressed by a number of *factored* HMMs such as *factorial* HMMs (Ghahramani & Jordan, 1997) and *mixed-memory* Markov models (Saul & Jordan, 1999). Here, the (hidden) state is factored into multiple state variables and is therefore represented in a distributed manner. Moreover, the distributed nature allows to devise an efficient variational approximation by (weakly) decoupling the state variables. The main contribution of the present extend abstract is to show how to lift this idea to the relational case.

2. Factored HMMs

Consider modeling string quartets. A violin has a pitch range from g till $a4$ this corresponds to four octaves each with 12 semi tones. Therefore, a string quartet can play $(4 \cdot 12)^4 \approx 5 \cdot 10^6$ combinations of notes (even more including double stops and flageolets). This number also corresponds to the required number of hidden state in an HMM modeling a string quartet. This is clearly an intractable state space. An

Extended abstract in the working notes of MLG-2007, August 1-3, Florence, Italy, 2007. Copyright 2007 by the author(s)/owner(s).

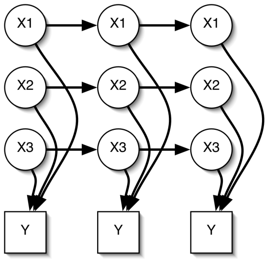

Figure 1. A factorial HMM: independent processes conspire to generate the observed output sequence.

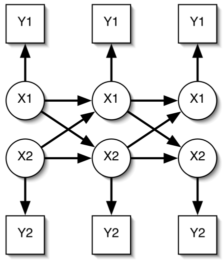

Figure 2. Two (weakly) coupled HMM: processes weakly interact to generate the independent output sequences.

alternative is to decompose the string quartet state space into four separate state variables, namely one for each instrument. This results in a much smaller number of states per state variable, namely only 48 values.

This decomposition is exactly the idea underlying factored HMMs. Figures 1 and 2 show two instances of factored HMMs, which represent extreme points of the factored HMM spectrum: factorial HMMs and coupled HMMs. Unfortunately, decomposing the state variables only does not make exact inference and learning algorithms tractable (Ghahramani & Jordan, 1997). The decomposition, however, paves the way for an approximative inference algorithm, which is cubic in the number of hidden state variables. The basic idea is that *each object (instrument) represented by a (hidden) state variable chooses its next state only based on the current joint state, i.e., independent of the next state of the other state variables.* This assumption together with making a structured mean field approximation allows us to show in the remainder of this extended abstract that the exponential runtime complexity drops from n^{2k} to k^3n^2 for one transition, where k is the number of random variables and n is the domain size of the random variables *even in the relational case.*

Why are we interested in the relational case? Reconsider our string quartet examples. The number of states for each (hidden) state variable is sill very high compared to the number of state variables: 48 vs. 4. So, why not factorizing even further? Well, decomposing the state for one instrument into two random variables – one for the note and one for the octave – we would encode that changing the semitone and the octave are independent of each other. Now assume that one instrument instrument transitions from the note 12^{th} one halftone up. The next note will be the first note but also one octave higher, which is wrong. Nevertheless, as we will argue in the next section, there are (context-specific) independencies among the state variables, which we would like to employ for fast inference.

3. Weakly-Coupled Relational HMMs

In a factored HMM, each hidden state consist of a vector of unstructured symbols. These symbols are represented by a set of random variables $\mathbf{X_t} = X_{1,t}, \ldots, X_{n,t}$. With the *term chain* we refer to the set $\bigcup_t X_{i,t}$ representing the same object over time. The random variables $X_{i,t}$ carries the information of the history over to the next state at time point $t+1$. As an example for a state consider:

$$\underbrace{note(basso,1,0)}_{X_{1,t}}, \underbrace{note(alto,1,1)}_{X_{2,t}}, \\ \underbrace{note(tenor,1,1)}_{X_{3,t}}, \underbrace{note(soprano,2,1)}_{X_{4,t}}$$

The state says that the instrument $note(basso,1,0)$ – represented by $X_{1,t}$ – plays the first note of the octave zero at time point t. We will call such a statement *ground state* and the combination of ground states for each $X_{i,t}$ at time t a *joint* ground state. Using ground states only, a traditional *factorial* HMMs requires to specify the conditional probability distribution (CPD) $P(X_{i,t+1}|X_{i,t})$ for each possible state value combination. Even in our simple examples this CPD consist of 2304 entries. Additionally the hidden state values can only depend via the output. For coupled HMMs, things get even worser. Now the number of parameters also depends on the number of hidden state variables. In our string quartet example, requires to specify roughly $255 \cdot 10^6$ parameters. This is clearly intractable. In contrast, relational HMMs allow to aggregate sets of ground states together by using logical atoms. For instance, $note(Voice, Note, Octave)$ refers to all ground states, in which an instrument $Voice$ plays a certain note $Note$ in a certain octave $Octave$. This abstraction in turn allows to compactly encode the probabilistic information. In the following, we will extend relational HMMs to the weakly-coupled case.

Weakly-coupled relational HMMs are the factored variant of logical HMMs (Kersting et al., 2006). Consequently, the state of the system at each time step is a

$$\text{abstract state} \begin{cases} \text{body:} & note(Voice, Note, Octave) \\ \text{guard:} & note(Other, Note, Octave2) \wedge Other \neq Voice. \end{cases}$$

$$\text{head:} \qquad \rightarrow note(Voice, Note, Octave2)$$

Figure 3. An abstract transition (probability value omitted) of a weakly-coupled relational HMM. Capitalized words denote placeholders (for ground properties of the state) to share knowledge across set of states by means of unification.

set of ground atoms (one for each chain) and not only a single ground atom. An abstract state consists of two components: a body (the state of a single chain) and a guard.

Definition 1 *An abstract state $\{B, \varphi\}$ consists of a body B and a guard φ. A body is a logical atom and specifies a set of ground atoms, i.e., concrete states for a chain. A mapping θ_B of the variables (placeholders) in B to objects in the domain (constants) instantiates the abstract state B to a ground state. The guard is a conjunction of logical atoms. It describes how one object is related to other objects in a state.*

Thus, whereas the body corresponds to an abstract state in the sense of relational HMMs (Kersting et al., 2006) and in turn specifies the properties of states of a *single* random variable, the guard defines properties and relation among all random variables. As we will see below, an abstract transition fires only if the guard is true. This can always be checked as the systems is at each time in exactly one state, i.e., one ground atom per chain. To break ties among matching abstract states, we assume the set of abstract states to be totally ordered according to some order.

As an example, consider the abstract state shown in Fig. 3. Its meaning is that two different instruments (voices) play the same note. First, the body says that there is a voice, which is playing some note. Then, the guard makes sure that there is no other voice playing the same note. Note that we assume that the system is at each in time in a particular joint ground state, i.e., we can match each placeholder (such as $Voice$, $Other$, etc.) to a domain element (constant). This variable mapping can in turn be used to specify a probability distribution over the next states, i.e., over the states the system can transition to. Following Kersting et al. (2006), we specify a distribution over possible successor states as follows.

Definition 2 *An abstract transition is an expression of the following form: $p :: \{B, \varphi\} \rightarrow H$ where p is a probability value, $\{B, \varphi\}$ denotes an abstract states, and H is a logical atom. An abstract transition belongs to exactly one abstract state only. This guarantees that only one abstract state can fire at each time for each state variable,i.e., chain. Furthermore, note that the variables appearing in the guard can be used in the head. In this way, we can share knowledge across individual chains.*

Figure 3 shows an example for an abstract transition. It states that the instrument playing $Voice$ takes over the octave of the another instrument (if the guard is true in the current joint state). If there are multiple true groundings of the guard, as $Other = soprano$ and $Other = alto$ when determining the abstract state for X_1 in the example, we select uniformally among them. Multiple successor states, i.e., free variables in the head are dealt with in the same way as for logical HMMs, namely by assume a selection disrtibution μ mapping atoms to ground atoms.

The only thing left is the definition of the *sensor* model, i.e., the probability model for making observations. To do so, we encode observations using a totally ordered set of expression of the form $p :: S_1, \ldots, S_m \rightarrow O$ where the S_i and O are logical atoms. Each time such a rule fires in a joint ground state, we make the corresponding observation (grouding free variables using the selection distribution μ).

Putting everything together, a weakly-couple, relational HMM consists of set of abstract states, transitions, and observations. It can be proven a long the lines of (Kersting et al., 2006) that each weakly-coupled relational HMM induces a unique probability distribution.

4. Structured Mean Field Approx.

Mean field theory provides an alternative perspective on inference. The intuition behind mean field is that in dense graphs each node is subject to influences from many other nodes. Assuming that each influence is rather weak and that the total influence is roughly additive, the law of large number suggest that each node should be roughly characterized by its mean value. Indeed, the mean value is unknown, but it is related to the mean values of the other nodes. For Bayesian networks and HMMs, it has been found that the mean value of a given node is obtained additively from the mean values of the nodes in its Markov blanket (Saul & Jordan, 1996).

For weakly-coupled HMMs, however, we can do even better. Each chain individually is tractable. Thus, we can improve the mean field approximation by decoupling only the variables across the chains. This is

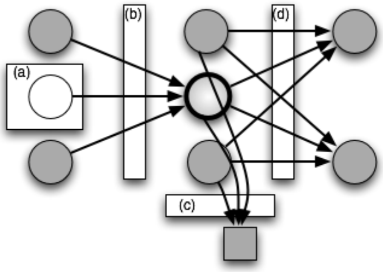

Figure 4. Information considered by the chainwise Viterbi algorithm to compute a transition probability: (a) the probability to reach the current state, (b) the transition probability of chain l, (c) the observation probability, (d) the transition probability of the other chain from t to $t+1$ given that chain l is at t in x_i.

called a *structured mean field* approach. Whenever the chains are only loosely coupled, we would expect this approximation to be quite accurate.

This basically leads to relational variants of Saul and Jordan (1999)'s *chain-wise* inference procedures for mixed-memory Markov models, which all follow the same principle and are akin to the hard EM. Let us illustrate this for the Viterbi algorithm, i.e., for computing the most-likely joint state sequence $\overline{x}_{i,1:T}$ given a sequence of observations $\overline{o}_{1:T}$. First, an initial guess is made for the Viterbi path $\overline{x}^{(0)}_{i,1:T}$ of each component relational HMM i, for instance by running the Viterbi algorithm for logical HMMs for each chains separately ignoring the inter-chain dependencies. Then, a *chain-wise* Viterbi algorithm is applied, in turn, to each of the relational HMMs. The chainwise Viterbi computes the optimal path of hidden $\overline{x}^{(l)}_{i,1:t}$ states through the ith chain given fixed values $\overline{x}^{(l-1)}_{i,1:t}$ of the last iteration for the hidden states of the other chains. This is essentially again the Viterbi algorithm for logical HMMs but it uses a modified transition probability:

$$
\begin{aligned}
&\delta(x^{(l)}_{i,t}|o_{1:t}) = \\
&\qquad \max\nolimits_{x_{i,t-1}} \delta(x^{(l)}_{i,t-1}|o_{1:t-1}) && \text{(a)}\\
&\qquad P(x^{(l)}_{i,t}|\overline{x}^{(l-1)}_{1:i-1,t-1}, x^{(l)}_{i,t-1}, \overline{x}^{(l-1)}_{i+1:n,t-1}) && \text{(b)}\\
&\qquad P(o_t|\overline{x}^{(l-1)}_{1:i-1,t}, x^{(l)}_{i,t}, \overline{x}^{(l-1)}_{i+1:n,t}) && \text{(c)}\\
&\qquad \prod_{j=1:n\setminus i} P(\overline{x}^{(l-1)}_{j,t+1}|\overline{x}^{(l-1)}_{1:i-1,t}, x^{(l)}_{i,t}, \overline{x}^{(l-1)}_{i+1:n,t}) && \text{(d)}
\end{aligned}
$$

As exemplified in Figure 4 , it takes the following probabilities into account: (a) the probability to reach the current state, (b) the transition probability of chain l, (c) the observation probability, and (d) the transition probability of the other chain from t to $t+1$ given that chain l is at t in x_i. After the chainwise Viterbi has been applied once to each chain, we iterate the cycle until convergence. With this algorithm one complete cycle can be computet in time k^3n^2 instead of the orginal n^{2k}.

5. Conclusions

We introduced weakly coupled relational HMMs (WCRHMMs). Based on a distributed, abstract state representation, we then developed a mean field approximation for efficient, approximative inference. Preliminary experiments have shown that the approximation works well in practice.

To the best of our knowledge, the inference procedure is the first application of a variational method within SRL. Investigating this connection for other SRL approaches is an interesting direction for future research as it paves the way towards general *relational, variational Bayes* methods.

Acknowledgments The authors thank Luc De Raedt for his support.

References

Anderson, C., Domingos, P., & Weld, D. (2002). Relational Markov Models and their Application to Adaptive Web Navigation. *Proc. of the 8th Int. Conf. on Knowledge Discovery and Data Mining (KDD-02)* (pp. 143–152).

Bengio, Y., & Frasconi, P. (1995). An input output HMM architecture. *Advances in Neural Information Processing Systems* (pp. 427–434). The MIT Press.

Ghahramani, Z., & Jordan, M. (1997). Factorial hidden Markov models. *Machine Learning Journal*, *29*, 245–273.

Jaakkola, T. (2000). Tutorial on variational approximation methods.

Kersting, K., De Raedt, L., & Raiko, T. (2006). Logial Hidden Markov Models. *Journal of Artificial Intelligence Research (JAIR)*, *25*, 425–456.

Rabiner, L. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, *77*, 257–286.

Sanghai, S., Domingos, P., & Weld, D. (2003). Dynamic probabilistic relational models. *Proc. of the 8th Int. Joint Conference on Artificial Intelligence (IJCAI-03)* (pp. 992–997).

Saul, L. K., & Jordan, M. I. (1996). Exploiting tractable substructures in intractable networks. *Advances in Neural Information Processing Systems* (pp. 486–492). The MIT Press.

Saul, L. K., & Jordan, M. I. (1999). Mixed memory markov models: Decomposing complex stochastic processes as mixtures of simpler ones. *Mach. Learn.*, *37*, 75–87.