Distributed Relational State Representations for Complex Stochastic Processes *

Ingo Thon[1] and Kristian Kersting[2]

[1] Katholieke Universiteit Leuven, Department of Computer Science
Celistijnenlaan 200A, 3001 Heverlee, Belgium
ingo.thon@cs.kuleuven.be
[2] Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory, 32 Vassar St, Cambridge, MA 02139, USA
kersting@csail.mit.edu

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 $O(n^k)$ to $O(n)$.

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 another relation. The algorithmic complexity for general purpose, dynamic SRL approaches easily explodes and becomes intractable in practice if quite strong assumptions are not 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 models, which naturally restrict the dynamics of the domain, to relational models.

* An earlier version of this work appeared as 4-pages extended abstract in the electronic working notes of the 5th International Workshop on Mining and Learning with Graphs (MLG'07), August 1–3, 2007, Universita degli Studi di Firenze, Florence, Tuscany, Italy. In the present paper, we report for the first time on experimental results.

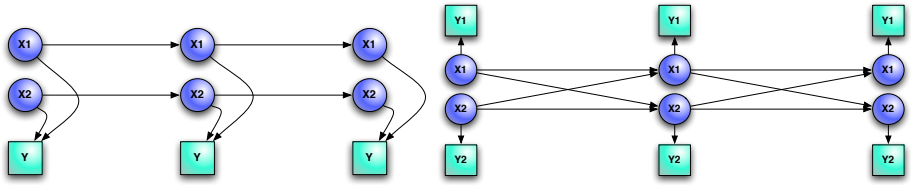


Fig. 1. Factored HMMs. (Left) a factorial HMM: independent processes X_i (ovals) are coupled through a single, joint output sequence Y_i (boxes). (Right) Weakly coupled HMMs: processes X_i (ovals) weakly interact to generate independent output sequences Y_i (boxes), one for each process.

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) itself are extremely popular for modeling dynamic domains. Application areas include computational biology, user modeling, 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 combinatorial 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 work is to show how to lift this idea to the relational case.

We proceed as follows. After briefly reviewing factored HMMs in the next Section, we will introduce weakly-couple relational HMMs (wcrHMMs) in Section 3. Section 4 then presents a structured mean field approximation for efficient inference within wcrHMMs. Before concluding, we will experimentally evaluate this approach.

2 Factored Hidden Markov Models

Consider modeling string quartets. A violin has a pitch range from g until $a4$ this corresponds to four octaves denoted by the number, each with 12 semi tones denoted by a letter and an optional modifier. For example, g corresponds to the 8th note of the zeros octave. 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 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 1 show two instances of factored HMMs, which represent extreme points of the factored HMM spectrum: factorial HMMs and coupled HMMs. Unfortunately, only decomposing the state variables 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 text that the exponential runtime complexity drops from $O(n^{2k})$ to $O(k^3n^2)$ 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 still 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 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 Hidden Markov Models

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

$$\underbrace{basso\_1\_0}_{X_{1,t}}, \underbrace{alto\_1\_1}_{X_{2,t}}, \underbrace{tenor\_1\_1}_{X_{3,t}}, \underbrace{soprano\_2\_1}_{X_{4,t}}$$

The state says that the instrument *basso* plays the first note of the octave zero at time point t represented by $X_{1,t}$. 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

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

Fig. 2. 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.

entries. Additionally the hidden state values can only depend via the output. For coupled HMMs, things get even more worse. Now the number of parameters also grows exponential in the number of chains. Our string quartet example, would require to specify roughly $2.5 \cdot 10^8$ parameters. This is clearly intractable.

In contrast, relational HMMs allow to aggregate sets of ground states together by using logical atoms. The above example rewritten in logical notation would be

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

Now for instance, $note(Voice, Note, Octave)$ refers to all ground states, in which an instrument $Voice$ plays any note $Note$ in any octave $Octave$. Where capitalized words denote variables and ground states are states where every variable is replaced by a constant value. 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 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 the set of all subsumed ground 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. The guard applied to a joint state also induces one or more mappings* $\theta_{\varphi,i}$*.*

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 relations, 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 arbitrary order.

As an example, consider the abstract state shown in Fig. 2. Its meaning is that two different instruments ($Voice$s) 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 another voice playing the same note. Note that we assume that the system is at each point 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 transits 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 state, and H is a logical atom. An abstract transition belongs to exactly one abstract state. Note that the variables appearing in the body and the guard can be used in the head. In this way, we can share knowledge across individual chains.

Figure 2 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 uniformly 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 assuming a selection distribution $\mu(a|A)$ mapping atoms A to ground atoms a.

Definition 3. *A* **selection distribution** $\mu(a|A)$ *defines for every logical atom A and every ground atom a the probability that a will be a ground instance of A.*

Additionally to the transition distribution there has to be a way to define the prior distribution π over the joint ground states. To this end, we assume a finite set of expressions of the form $p :: \{H_1, \ldots, H_n\}$, i.e., one atom H_i per chain. Then, using the selection distribution, we define

$$\pi(\{h_1, \ldots, h_n\}) = P(\mathbf{X_0} = \{h_1, \ldots, h_n\}) = \alpha \sum\nolimits_{p::\{H_1,\ldots,H_n\}\in\pi} p \cdot \prod\nolimits_{i=1}^{n} \mu(h_i|H_i)$$

where α is a normalization constant. Note, however, that this is not the only way one can imagine to specify a prior over joint ground states and any of them will work fine with our inference procedure we will introduce below.

The only thing left is the definition of the *sensor* model, i.e., the probability model for making observations.

Definition 4. *A sensor model is a set of expressions of the form*

$$p \ :: S_1, \ldots, S_m \rightarrow O$$

where the S_i and O are logical atoms.

Each time an observation rule fires (assuming the same conflict resolution rule as for abstract transition rules) in a joint ground state, we make the corresponding observation (grouding free variables using the selection distribution μ).

Putting everything together results into the definition of a weakly-coupled relational HMMs.

Definition 5. *A* **weakly-coupled relational HMM** *(*WCRHMM*) consists of a set of totally ordered abstract states, sets of abstract transitions for every abstract state, a selection distribution μ, a initial state distribution π, a set of observations.*

A long the lines of Kersting et al. (2006), one can prove that every WCRHMM defines a unique probability distribution.

Theorem 1. *A weakly-coupled relational HMM defines a time discrete stochastic process $\langle \mathbf{X}_t \rangle_t$. The induced probability measure over the Cartesian product over all random variables exists and is unique for each $t > 0$ and in the limit $t \to \infty$.*

To see this, note that every WCRHMM with a finite number of chains can be translated into a logical HMM: one basically computes the Cartesian product of all abstract states and the resulting abstract transitions.

The proof of Theorem 1 provides us with a general way to do inference and learning within WCRHMMs: compile the WCRHMM into a logical HMMs and use the inference techniques developed for logical HMMs (Kersting & Raiko, 2005; Kersting et al., 2006). This approach, however, typically scales as n^2, where n is the number of hidden state. In practice, exact inference is therefore limited to relational HMMs with relative small state spaces.

4 Structured Mean Field Approximation

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 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

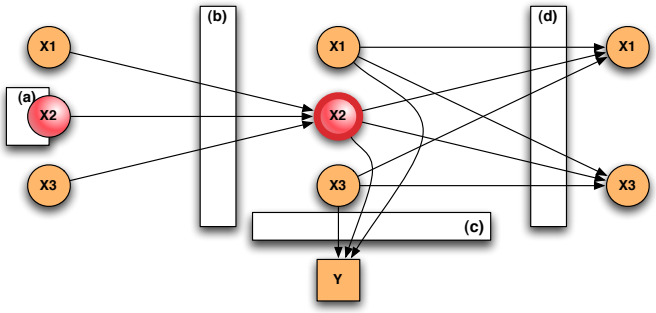


Fig. 3. Probabilistic information employed by the chainwise Viterbi algorithm to compute a transition probability for chain having all other chain fixed: (a) the probability to reach the last state, (b) the transition probability of chain i, (c) the observation probability, (d) the transition probability of the other chain from t to $t+1$ given that chain i is at t in x_i.

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 $\overline{x}_{i,0:T}$ given a sequence of observations $o_{1:T}$. First, an initial guess is made for the Viterbi path $\overline{x}^{(0)}_{i,0:T}$ 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. This is done by ignoring the guard. Then, a *chainwise* Viterbi algorithm is applied, in turn, to each of the relational HMMs. The chainwise Viterbi computes the optimal path of hidden $\overline{x}^{(l)}_{i,0:t}$ states through the ith chain given fixed values $\overline{x}^{(l-1)}_{j,0:t}$ 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:

$$\delta(x^{(l)}_{i,t}|o_{1:t}) = \max_{x_{i,t-1}} \delta(x^{(l)}_{i,t-1}|o_{1:t-1}) \quad \text{(a)}$$

$$P(x^{(l)}_{i,t}|\overline{x}^{(l-1)}_{1:i-1,t-1}, x^{(l)}_{i,t-1}, \overline{x}^{(l-1)}_{i+1:n,t-1}) \quad \text{(b)}$$

$$P(o_t|\overline{x}^{(l-1)}_{1:i-1,t}, x^{(l)}_{i,t}, \overline{x}^{(l-1)}_{i+1:n,t}) \quad \text{(c)}$$

$$\prod_{j=1:n\setminus i} P(\overline{x}^{(l-1)}_{j,t+1}|\overline{x}^{(l-1)}_{1:i-1,t}, x^{(l)}_{i,t}, \overline{x}^{(l-1)}_{i+1:n,t}) \quad \text{(d)}$$

where, cf. Figure 3, (a) is the probability to reach the last state, (b) is the transition probability of chain i, (c) is the observation probability, and (d) is the transition probability of the other chain from t to $t+1$ given that chain i is at t in $x^{(l)}_{i,t}$. After the chainwise Viterbi has been applied once to each chain, we iterate the cycle until convergence. The complete procedure RCVITERBI is given in Algorithm 1. One complete cycle of Algorithm 1 can be computed in time $O(k^3n^2)$ instead of the original $O(n^{2k})$.

Algorithm 1 RCVITERBI: Relational chainwise Viterbi

1: **procedure** UPDATE-PATH($x_{0:T}$, $o_{1:T}$, $\overline{x}_{0:T}$, i)
2: $p_{x,0} \leftarrow \pi(\overline{x}_{1,0} \dots \overline{x}_{1,i-1}, x, \overline{x}_{0,i+1} \dots \overline{x}_{0,n})$ ▷ init $p_{x,0}$. In general, $p_{x,i}$ stores the probability of the most likely path for $o_{1:t}$, which ends in x.
3: **for all** $t \in [1 \dots T]$ **do**
4: **for all** x **do**
5: $p_{x,t} \leftarrow 0$ ▷ init $p_{x,t}$, i.e., the probability of being in x at t
6: **end for**
7: **for all** x' with $p_{x',t-1} > 0$ **do** ▷ Consider only states reachable at $t-1$
8: $\{B, \varphi\} \leftarrow$ abstract state matching $\overline{x}_{1,t-1} \dots \overline{x}_{1,i-1}, x', \overline{x}_{t-1,i+1} \dots \overline{x}_{t-1,n}$
9: $\theta_B \leftarrow$ mgu of x' and B ▷ Ground the variables in the body
10: **for all** θ_φ s.t. $\varphi\theta_B\theta_H$ contains no free variable and is true in state $\overline{x}_{1,t-1} \dots \overline{x}_{1,i-1}, x', \overline{x}_{t-1,i+1} \dots \overline{x}_{t-1,n}$ **do**
11: **for all** $p :: \{B, \varphi\} \rightarrow H$ **do** ▷ For all abstract successors of x'
12: $p_{new} \leftarrow 0$
13: **for all** groundings x of $H\theta_B\theta_\varphi$ **do** ▷ For all ground successors, compute modified transition probilities (lines 13 – 21)
14: $p_a \leftarrow p_{x',t-1}$
15: $p_b \leftarrow p \cdot \mu(x|A)$
16: $p_c \leftarrow 0$
17: **for all** $p_O :: S_1, \dots, S_m \rightarrow O$ s.t. $S_1 \dots, S_m$ is true in $\overline{x}_{1,t-1} \dots \overline{x}_{1,i-1}, x', \overline{x}_{t-1,i+1} \dots \overline{x}_{t-1,n}$ **do**
18: $p_c \leftarrow p_c + p_O\mu(o, O)$
19: **end for**
20: $p_d \leftarrow 1$
21: **for all** $j < n$ and $j \neq i$ **do**
22: $p_d \leftarrow p_d \cdot P(x_{t+1,j}|\overline{x}_{1,t} \dots \overline{x}_{1,i-1}, x, \overline{x}_{t,i+1} \dots \overline{x}_{t,n})$
23: **end for**
24: $p_{new} \leftarrow p_{new} \cdot p_a \cdot p_b \cdot p_c \cdot p_d$
25: **if** $p_{new} > p_{x,t}$ **then** ▷ If more likely, set as current best path
26: $p_{x,t} \leftarrow p_{new}$
27: $pred(x, t) \leftarrow x'$
28: **end if**
29: **end for**
30: **end for**
31: **end for**
32: **end for**
33: **end for**
34: $x \leftarrow \arg\max_x p_{x,T}$ ▷ Extract the computed Viterbi path.
35: **for** t=T-1...0 **do**
36: $\overline{x}_{i,t} \leftarrow pred(x, t)$
37: $x \leftarrow \overline{x}_{i,t}$
38: **end for**
39: **end procedure**

5 Experimental Demonstration

To demonstrate the relational chainwise Viterbi algorithm, consider the vacuum world of Russell and Norvig (1995) as depicted in Figure 4 for the case of 4 rooms.

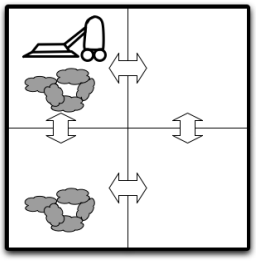


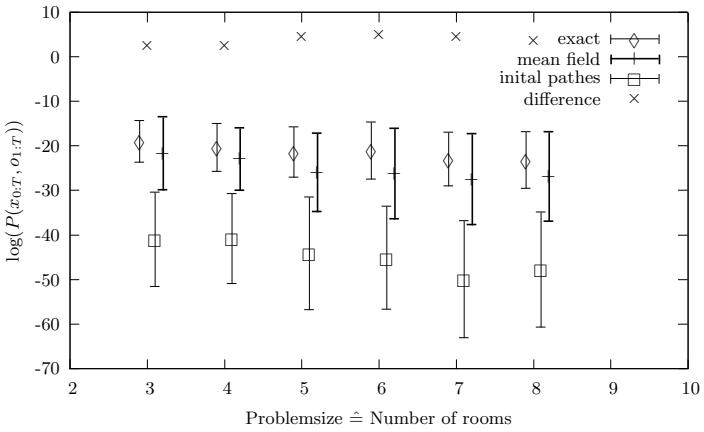
Fig. 4. Illustration of the Vacuum world we used to demonstrate RCVITERBI. Here we assume 4 rooms which are arranged in a circle. The robot is in the upper-left room.

Example 1. In the Vacuum world, there are n rooms and a single robot. The robot has two actions to choose from: walking (w) and cleaning (c). Rooms `X` and `Y` are connected via `door(X,Y)`, which the robot can use to walk from `X` to `Y`. If the robot is in a `Room` and cleaning, the room will be clean (`clean(Room)`) with a chance of 90% after the cleaning action. A clean room will stay clean in any case and a dirty room (`dirty(Room)`) will also stay dirty by default if not performing the cleaning action. The action the robot chooses correspond with probability 0.8 to the true state of the current room. In other words, the robot is not always able to determine the current state of the room correctly. The observation at every time consists of the position of the robot and dirt level of the room. This information is only with a probability 0.75 correct. In the other cases either the position of the robot is wrong or the dirt level or both.

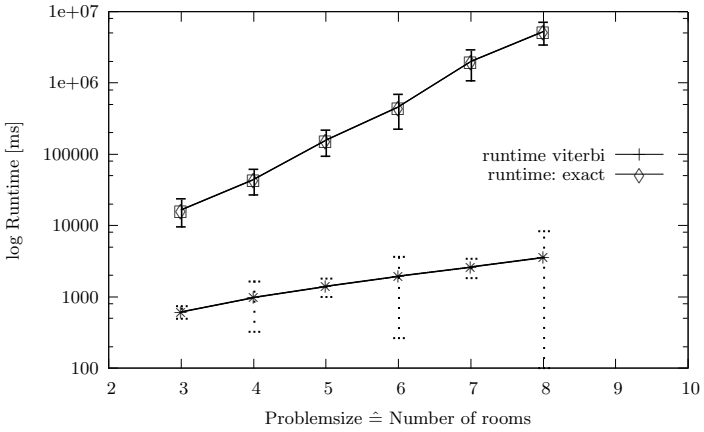
To model the Vacuum world as a WCRHMM, we treated `robot(Room,Action)`, `clean(Room)`, and `dirty(Room)` as abstract chain. The rooms as well as the topological information among them, i.e., `door(X,Y)`, is provided apriori as deterministic background knowledge. Then the Vacuum world can be modeled as follows:

$$
\begin{array}{lll}
0.9 :: & clean(X) \leftarrow dirty(X) & \{robot(X,c)\} \\
0.1 :: & dirty(X) \leftarrow dirty(X) & \{robot(X,c)\} \\
1.0 :: & dirty(X) \leftarrow dirty(X) & \{\} \\
1.0 :: & clean(X) \leftarrow clean(X) & \{\} \\
0.8 :: & robot(X,c) \leftarrow robot(X,\_) & \{door(X,Y) \wedge dirty(X)\} \\
0.2 :: & robot(Y,w) \leftarrow robot(X,\_) & \{door(X,Y) \wedge dirty(X)\} \\
0.2 :: & robot(X,c) \leftarrow robot(X,\_) & \{door(X,Y) \wedge \neg dirty(X)\} \\
0.8 :: & robot(Y,w) \leftarrow robot(X,\_) & \{door(X,Y) \wedge \neg dirty(X)\}
\end{array}
$$

Based on this model, we compared the exact and the chainwise relational Viterbi algorithms. More precisely, for an increasing number of rooms $(3, 4, \ldots, 8)$, we randomly sampled 40 observation sequences of length 10. For each sequence we then ran both algorithms to compute the most-likely hidden state sequence. The mean field approach was set to spend 4 iterations per element in the interpretation.



(a) Qualitative comparison: Number of rooms (x axis) vs. log joint probability $log(P(x_{0:T}, o_{1:T}))$ of computed Viterbi path and observation sequence (y axis), which is bascially maximized by the Viterbi algorithm. Finally, we show the probabilities of the paths, which had been used to initialize the mean field approximation. The mean of the latter probabilities roughly corresponds to the expected probability of a randomly selected state sequence.



(b) Running time comparison: Number of rooms (x axis) vs. runtime (y axis) in ms on a logarithmic scale. The skewness of error bars are due to the logarithmic scale

Fig. 5. Experimental results on the vacuum world domain averaged over 40 sequences: (a) qualitative comparison, (b) running time comparison. The results show that the structured mean field approximation achieves a performance, which is competitive with the exact inference approach, but it is several orders of magnitude faster.

The experimental results are summarized in Figures 5(a) and 5(b). As one can see in Figure 5(a), the chainwise Viterbi approach yields close approximations

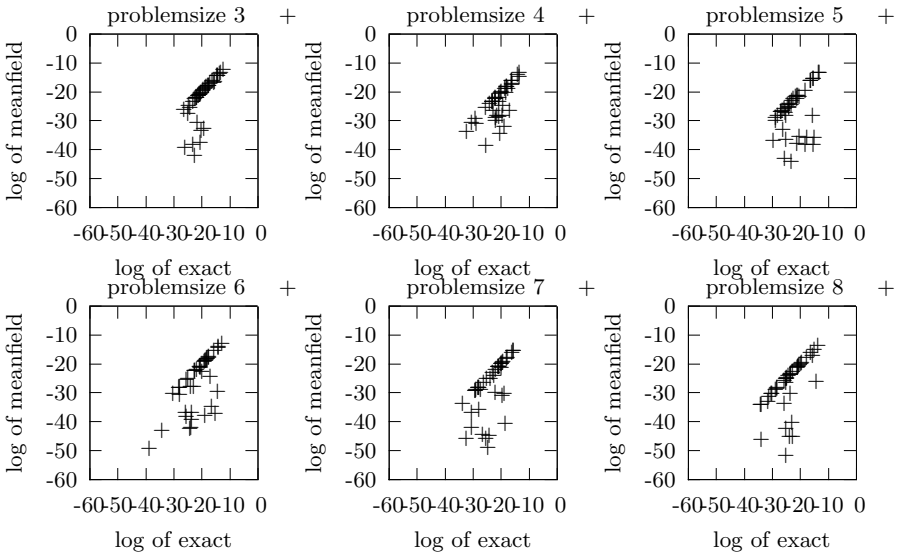


Fig. 6. Scatter-plots of the 40 experiment made for every problemsize. This shows that the results are in most cases equally good. In the other cases the solutions are still reasonable.

of the true probabilities. It is, however, an order of magnitude faster, cf. Figure 5(b), as predicted by the theory: the exact viterbi algorithm is exponential in the number of rooms as the number of possible hidden states grows exponential in the number of rooms. The quantitative results were as follows:

	#rooms					
# of paths with	**3**	**4**	**5**	**6**	**7**	**8**
same probabilities	28	18	17	20	22	27
$0\% <$ log range $\leq 5\%$	1	3	3	3	3	3
$5\% <$ log range $\leq 10\%$	3	6	7	1	3	1
subtotal (absolute/relative)	**32/.8**	**27/.67**	**27/.67**	**24/.6**	**28/.7**	**31/.77**
$> 10\%$ log range	8	13	13	16	12	9
total	**40**	**40**	**40**	**40**	**40**	**40**

Thus, in most cases both algorithms output a path with the same probability. In the cases in which the estimated path RCVITERBI is suboptimal, the solution is still reasonable.

This is also illustrated in the scatter-plots in Figure 6.

To summarize the experiments demonstrate that RCVITERBI achieves comparable performance as the exact approach but is several orders of magnitudes faster.

6 Conclusions

We introduced weakly coupled relational HMMs (wcrHMMs). Based on a distributed, abstract state representation, we then developed a structured mean field approximation for efficient, approximative inference. First experiments have shown that the approximation works well in practice. This experiments have also shown, that the exact algorithm is intractable even in the simple cases, because of the exponential growth of the runtime in the size of the interpretations. 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. The research was partly supported by the Research Foundation-Flanders (FWO-Vlaanderen).

Bibliography

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).

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

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

Kersting, K., & Raiko, T. (2005). 'Say EM' for Selecting Probabilistic Models for Logical Sequences. *Proc. of the 21st Conf. on Uncertainty in Artificial Intelligence (UAI-05)* (pp. 300–307).

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

Russell, S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Inc.

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.