Factorial Hidden Markov Models

Emile Mathieu
Department of Mathematics and Computer Science
Ecole des Ponts ParisTech
emile.mathieu.@eleves.enpc.fr

Abstract

This project consists in the study of the article [1] written by Ghahramani and Jordan, who constructed efficient learning algorithms for hidden Markov models with distributed state representations.

1 Introduction

The hidden Markov model (HMM) is the most used tool for discrete time-series modelling, and has applications ranging from speech recognition to computational molecular biology. HMM is a graphical model composed by a Markov chain with unobserved (hidden) states, and observed outputs which are each characterized by a conditional distribution given their associated hidden state. The Baum-Welch algorithm is an efficient algorithm for HMM's parameters estimation. This algorithm is tractable because of the multimodal assumption over state variables.

This assumption limits the representation capacity of HMM, and that is why distributed state representation have been considered by Williams and Hinton [3]. Such a representation can be preferable since the model can automatically decompose the state space into features, and because a priori information about the process' generation may be used. Moreover, an HMM with a distributed state representation can represent n bits of information with n binary state variables, whereas an HMM would need $K = 2^n$ distinct states. In [1], authors present efficient learning algorithms for factorial HMM (FHMM): an HMM with a distributed state representation.

The work presented below aims at describing in a succinct manner the probabilistic model of factorial hidden Markov models along with several EM algorithms tackling its associated parameters learning problem. Most algorithms have been reimplemented in Matlab and ran on simulated data and on J.S. Bach's chorales so as to assess their performances both in time complexity and likelihood of obtained models.

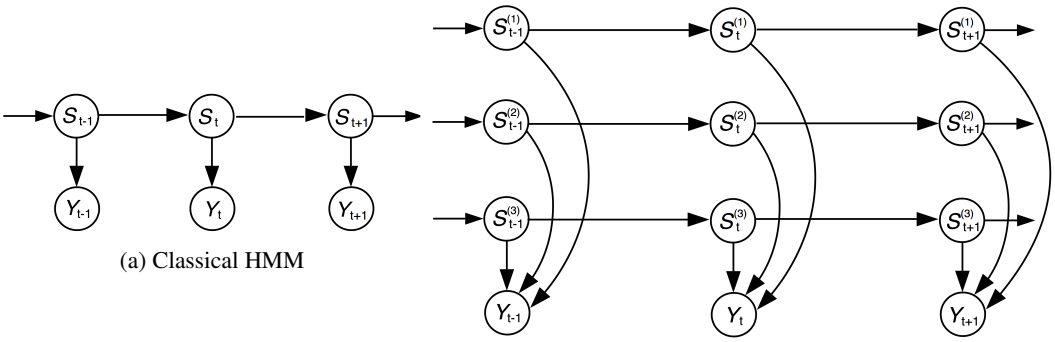
This report is organized in the following way: First, we recall the probabilistic model of the classical hidden Markov model and present its extension: the factorial hidden Markov model. Then we highlight the intrinsic difficulty of the parameters learning task with an exact EM algorithm. Next, we describe sampling based and variational approximations of the E step so as to propose scalable algorithms. After, we run those algorithms on synthetic and real data in order to assess their performances. Finally, we present an extension of factorial hidden Markov model and conclude.

2 Probabilistic models

2.1 Classical hidden Markov model

In the hidden Markov model, a sequence of observations $\{Y_t\} = \{Y_1, \dots, Y_T\}$ is modeled by a specific probabilistic relation to a sequence of hidden states $\{S_t\}$, itself modeled as a Markov chain. The two sets of conditional independence relations made in this model are the following: S_t is independent of $\{S_1, \dots, S_{t-2}\}$ given S_{t-1} (and of $\{S_{t+2}, \dots, S_T\}$ given S_{t+1}) because of the Markov property, and Y_t is independent of all other variables given the associate hidden state S_t. These relations are represented by a probabilistic graphical model in Figure 1a. The joint probability of the model can then be factored as:

$$P(\{S_t, Y_t\}) = P(S_1)P(Y_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t) \tag{1}$$



(a) Classical HMM

(b) Factorial HMM

Figure 1: DAGs representing conditional independence relations in HMMs

2.2 Factorial hidden Markov model

Instead of considering a unique Markov chain for the state variables as in HMM, factorial HMM (FHMM) represents the state by a collection of M independent Markov chains, as shown in Figure 1b. These Markov chains are independent, and generally have different transitions matrices $P^{(m)}$ and initial probabilities $\pi^{(m)}$, and take value in $\{1, \dots, K\}$. The state is now represented as a collection of states variables $S_t = \{S_t^{(1)}, \dots, S_t^{(M)}\}$ for each time step t. Since the Markov chains are independent:

$$P(S_t|S_{t-1}) = \prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)})$$

At time t, observation Y_t can depend on all states variables $S_t = \{S_t^{(1)}, \dots, S_t^{(M)}\}$. The joint probability can finally be factored as:

$$P(\{S_t, Y_t\}) = P(S_1)P(Y_1|S_1)\prod_{t=2}^{T} P(Y_t|S_t)\prod_{m=1}^{M} P(S_t^{(m)}|S_{t-1}^{(m)}) \tag{2}$$

In [1], authors consider a Gaussian model for continuous observations $\{Y_t\}$, whose mean is a linear function of the state variables:

$$P(Y_t|S_t) = |C|^{-\frac{1}{2}}(2\pi)^{-\frac{D}{2}} \exp\left\{-\frac{1}{2}(Y_t - \mu_t)' C^{-1}(Y_t - \mu_t)\right\}$$

$$\text{where } \mu_t = \sum_{m=1}^{M} W^{(m)} S_t^{(m)}$$

3 Learning

As for HMM, an EM algorithm is used to learn FHMM's parameters. The maximization step is simple and tractable. It can be derived by computing the gradient of the expectation of the complete log likelihood given the observations. Setting this gradient to zero yields closed form solutions for the parameters $\phi = \{W^{(m)}, \pi^{(m)}, P^{(m)}, C\}_{\{m=1,\dots,M\}}$. These formulas depend on the first and second order statistics: $\left\langle S_t^{(m)} \right\rangle$, $\left\langle S_{t-1}^{(m)} S_t^{(m)'} \right\rangle$ and $\left\langle S_t^{(m)} S_t^{(n)'} \right\rangle$, with $\langle \cdot \rangle = \mathbb{E}\left[\cdot | \phi, \{Y_t\}\right]$.

Those statistics shall be computed in the E-step, after having computed the posterior probabilities over the hidden state given parameters and observations. This step is the complicated part of the EM algorithm.

3.1 Exact

A forward-backward like algorithm that implements the exact E step exists, and is detailed in Appendix B of [1]. Unfortunately, this computation is intractable for large values of K and M because the algorithm has a complexity of $O(TK^{2M})$.

Indeed, even if the Markov property can be used to obtain a forward-backward-like factorizations of the expectations across time steps, one cannot avoid the summation over all possible configurations of other hidden state variables within each time step t.

3.2 Gibbs Sampling

In order to avoid to sum over exponentially many states, one can relate on a Monte Carlo sampling procedure. Gibbs sampling iteratively sample state variables from their conditional distribution which is relatively simple because each node is conditionally independent of all other nodes given its Markov blanket:

$$\begin{aligned} S_t^{(m)} &\sim P(S_t^{(m)} | \{S_t^{(n)} : n \neq m\}, S_{t-1}^{(m)}, S_{t+1}^{(m)}, Y_t, \phi) \\ &\propto P(S_t^{(m)} | S_{t-1}^{(m)}) P(S_{t+1}^{(m)} | S_t^{(m)}) P(Y_t | S_t^{(1)}, \dots, S_t^{(M)}) \end{aligned}$$

Under the assumption that all probabilities are bounded away from zero, the Markov chain is guaranteed to converge to the posterior probabilities of the states given the observations.

First and second-order statistics needed for the M-step are computed using the states visited during the sampling process.

3.3 Variational methods

Another class of approximation methods for inference are variational methods. The main idea of variational methods is to approximate the posterior distribution over the hidden variables $P(\{S_t\}|\{Y_t\})$ by a tractable parametrized distribution $Q(\{S_t\}|\theta)$ which does not depend on observations $\{Y_t\}$.

One theoretical advantage of variational methods over Gibbs sampling is that the variational approximation Q provides a lower bound on the log likelihood $P(\{S_t\})$. Indeed

$$
\begin{aligned}
\log p(\{S_t\}) &= \log \sum_{\{S_t\}} P(\{S_t, Y_t\}) \\
&= \log \sum_{\{S_t\}} Q(\{S_t\}) \frac{P(\{S_t, Y_t\})}{Q(\{S_t\})} \\
&= \mathrm{KL}(Q||P) + \sum_{\{S_t\}} Q(\{S_t\}) \log \frac{P(\{S_t, Y_t\})}{Q(\{S_t\})}
\end{aligned}
$$

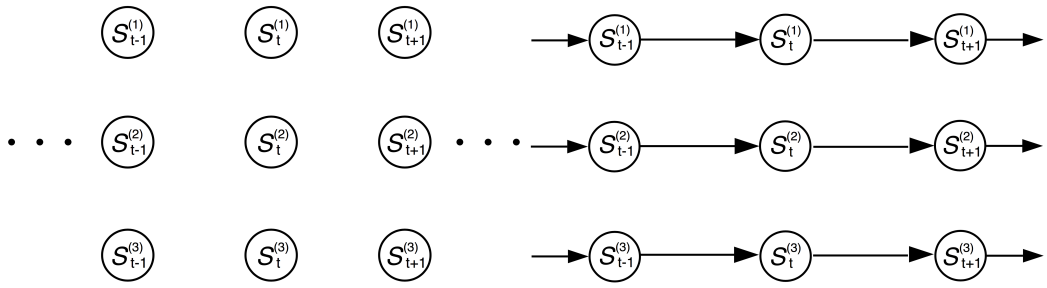
Once the structure of the variational distribution $Q(\{S_t\}|\theta)$ has been chosen, parameters θ can then tuned so as to obtain the tightest bound between $P(\{S_t\}|\{Y_t\})$ and $Q(\{S_t\}|\theta)$ by minimizing $\mathrm{KL}(Q||P)$.

3.3.1 Mean Field

The simplest variational approximation considers a completely factorized approximation where there is no more edge between $S_t^{(m)}$ state variables as shown in Figure 2a. Hence, this variational distribution can be factorized as:

$$
Q(\{S_t\}|\theta) = \prod_{t=1}^{T} \prod_{m=1}^{M} Q(S_t^{(m)}|\theta_t^{(m)})
$$

where parameters $\theta_t^{(m)}$ define the state occupation probabilities for the multinomial variable $S_t^{(m)}$ under the variational distribution Q.



(a) Completely factorized (b) Structured

Figure 2: DAGs representing the conditional independence relations in variational approximation models

Minimizing $\mathrm{KL}(Q||P)$ for this completely factorized distribution Q yields a set of fixed point equations of the form $\theta_t^{(m)\ new} = f(\theta_t^{(l\neq m)}, \theta_{t-1}^{(m)}, \theta_{t+1}^{(m)})$. Thus, parameters can be computed iteratively by updating $\theta_t^{(m)}$ for $m = 1 \ldots, M$ and $t = 1, \ldots, T$ until convergence. This convergence is monitored thanks to the KL divergence.

We observe that even if hidden variables are assumed to be independent, their mean are coupled since the fixed point equations couple the parameters associated with each node with the parameters of its Markov blanket. Information which used to be propagated in the exact algorithm, is here propagated by the fixed point equations.

3.3.2 Structured Mean field

The structured mean field method relaxes the extreme independence assumption of mean field. Within this scheme, FHMM is approximated by M uncoupled HMMs as shown in Figure 2b. Hence, this variational distribution can be factorized as:

$$
Q(\{S_t\}|\theta) = \prod_{m=1}^{M} Q(S_1^{(m)}|\theta) \prod_{t=2}^{T} Q(S_t^{(m)}|S_{t-1}^{(m)}, \theta)
$$

$$\text{with } Q(S_t^{(m)}|S_{t-1}^{(m)},\theta) = \prod_{k=1}^{K}\left(h_{t,k}^{(m)}\prod_{j=1}^{K}\left(P_{k,j}^{(m)}\right)^{S_{t-1,j}^{(m)}}\right)S_{t,k}^{(m)}$$

The parameters of the variational distribution are $\theta = \{\pi^{(m)}, P^{(m)}, h_t^{(m)}\}_{(m=1,\dots,M)}$, with $h_t^{(m)}$ being a time-varying bias for each state variable. It can be seen that $h_t^{(m)}$ plays the role of the emission probability $P(Y_t|S_t)$, when comparing previous equations with the joint probability for HMM.

As before, minimizing KL$(Q||P)$ yields a set of fixed point equations, here of the form $h_t^{(m)\ new} = g\left(\left\langle S_t^{(l)}\right\rangle\right)$.

Unfortunately, contrary to the completely factorized approximation where the fixed point equations was explicitly written in terms of the variational parameters, here these equations depend on $\left\langle S_t^{(m)}\right\rangle$. So the dependence of $\left\langle S_t^{(m)}\right\rangle$ on $h_t^{(m)}$ must be computed via the HMM's forward backward algorithm. Then $h_t^{(m)}$ is computed via the fixed point equation given $\left\langle S_t^{(m)}\right\rangle$. These two steps are repeated until convergence to solve the E-Step. This convergence is monitored thanks to the KL divergence.

4 Experiments

In this section, we apply the previously described algorithms on synthetic and real data so as to assess their performances both in term of computation time and likelihoods of obtained models.

4.1 Synthetic data

This experiment aims at comparing different approximate and exact methods of inference on a likelihood basis.

M	K	Algorithm	Training Set	Test Set
3	2	True	0.00 ± 0.21	0.00 ± 0.22
		HMM	0.85 ± 0.56	1.33 ± 0.54
		Exact	0.47 ± 0.98	0.56 ± 0.95
		SVA	0.57 ± 0.68	0.65 ± 0.66
3	3	True	0.00 ± 0.13	0.00 ± 0.13
		HMM	0.65 ± 0.48	7.18 ± 2.14
		Exact	0.83 ± 1.11	1.01 ± 1.05
		SVA	0.92 ± 1.09	1.15 ± 1.06
5	2	True	0.00 ± 0.37	0.00 ± 0.38
		HMM	0.40 ± 0.57	9.04 ± 4.18
		Exact	1.62 ± 1.12	1.76 ± 1.11
		SVA	1.94 ± 1.04	2.09 ± 1.08
5	3	True	0.00 ± 0.27	0.00 ± 0.28
		HMM	-3.32 ± 0.24	110.37 ± 21.59
		Exact	2.42 ± 1.66	2.66 ± 1.65
		SVA	2.82 ± 1.35	3.06 ± 1.41

Table 1: Comparison of inference algorithms

Synthetic data is generated from a FHMM. Parameters are uniformly sampled and normalized when necessary, except the covariance matrix which is set to $C = 0.0025I$. Several values of K and M are considered, and for each problem size, 15 sets of parameters are sampled. For each randomly sampled set of parameters, a separate training set and test set of 20 sequences of length 20 were

generated. Algorithms were run on those sets for a maximum of 100 iterations, and were stop early if the condition $L(k) - L(k-1) < 10^{-5}(L(k-1) - L(2))$ were satisfied, with $L(k)$ being the log likelihood at iteration k. Log likelihoods for each algorithm were computed and averaged over the 15 runs.

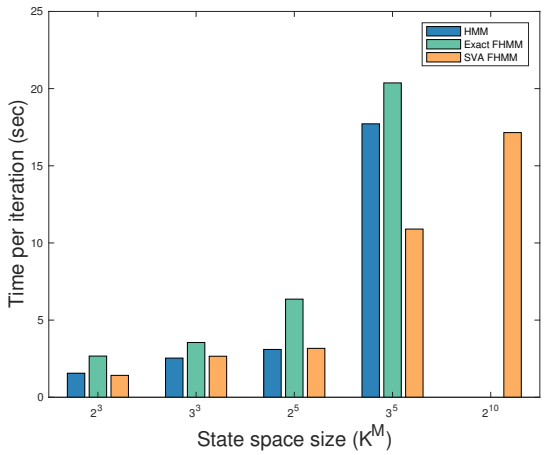


Figure 3: Time per iteration of several EM algorithms.

Results presented in Table 1 show that performances of the structured variational approximation are not significantly different from performances of the exact algorithm either on training datasets or on test datasets. Moreover, it appears that standard HMM tends to overfit since its test log likelihoods are significantly larger than training log likelihoods, even for the smallest state space size ($K = 2$ and $M = 3$). Not surprisingly, this overfitting issue grows with the size of the state space as highlighted in Table 1.

Theoretically, we anticipated that both the standard HMM and the exact E step factorial HMM would be extremely slow due to their complexity growing exponentially fast. In order to assess this prediction, we record execution times of EM algorithms for several values of K and M, and average them on 15 runs. Figure 3 represents times per iteration for standard HMM, exact E step FHMM and approximated E step with SVA. This figure shows that empirically standard HMM and the exact E step factorial HMM becomes very slow for large state spaces. Yet, the time per iteration for the variational methods scaled well to large state spaces.

4.2 Real data

This experiment intends to determine whether the decomposition of the state space in FHMMs can present any advantage.

The dataset consists of discrete event sequences encoding melody lines of J.S. Bach's Chorales, obtained from the UCI Repository for Machine Learning Databases ([2]). These musical pieces are expected to exhibit complex structures and that its "state" would come from the association of several latent features. Consequently, we hope that a factorial HMM could provide a more satisfactory model than a classical HMM for these musical pieces.

Sixty-six chorales truncated to 40 events each, were divided into a training set of 30 chorales and a test set of 36. The same stopping criterion than for synthetic data was used. HMMs with state space ranging from 2 to 100 states were trained until convergence. FHMMs of varying sizes were also trained on the same data with a structured variational approximation for the E step. Indeed, the structured variational approximation has similar empirical performances than the exact inference algorithm but was way faster in computational time for large state space size according to the previous section. Those test set log likelihoods are shown in Figure 4.

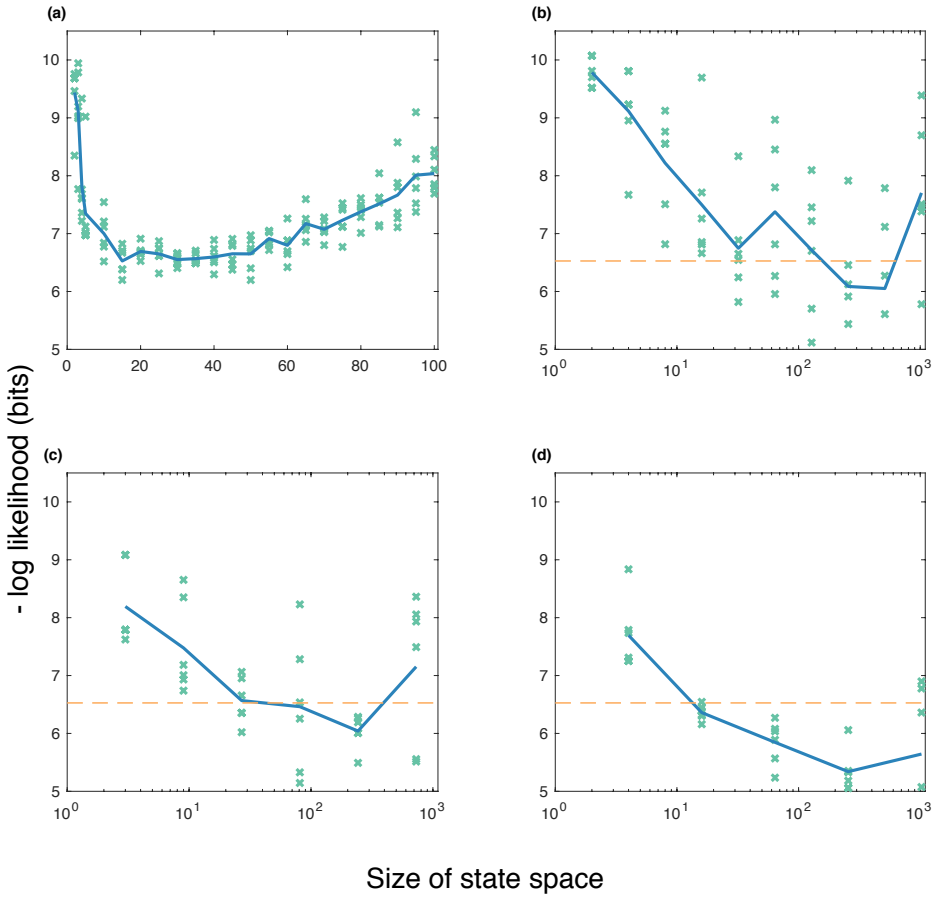


Figure 4: Test set log likelihoods per observation for HMM (a), FHMM with $K = 2$ (b), $K = 3$ (c) and $K = 4$ (d). Green crosses represent a single run and blue lines indicate the mean performances. The orange dashed line in (b-d) indicates the negative log-likelihood per observation of the best run in (a).

Log likelihoods presented in Figure 4 (a) display HMMs' inability to generalize training data with states bigger than 35. Indeed, the negative log-likelihood for the HMM exhibits a typical U shape, reaching the minimum value 6.56 bits for $K = 35$. This HMM's best performance is plotted as an orange dashed line in Figures 4 (b-d) to ease comparison between models.

It appears in Figures 4 (b-d) that factorial HMMs provide a richer model for J.S. Bach chorales than classical HMMs. Indeed, FHMMs are better predictors since better test log-likelihoods are observed such as for $K = 4$ and $M \geq 2$ (Figure 4 (d)) or for $K = 2$ and $8 \leq M \leq 9$ (Figure 4 (b)). Moreover, thanks to the low time complexity of the structure variational approximation, significantly larger state spaces can considered, up to 1000 in our experiments.

5 Extensions

In [1], Ghahramani and Jordan propose several generalizations of the factorial HMM model. The main one is the hidden Markov decision tree model represented in Figure 5. In this model all states variables $S_t^{(m)}$ depends on an input X_t and $S_t^{(n)}$ for $1 \leq n < m$. For each time step t, the architecture can be seen as a probabilistic decision tree with decision variables $S_t^{(m)}$. For a given value of the input X_t, the top node $S_t^{(1)}$ partitions the decision space into K decision regions. Then next nodes $\{S_t^{(2)}, \ldots, S_t^{(M)}\}$ repeatedly subdivide each of the previous regions into subregions.

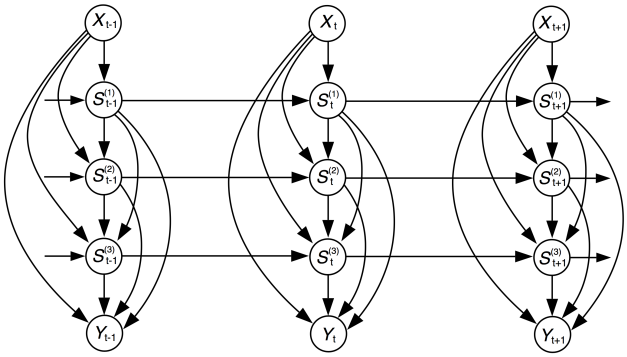


Figure 5: A DAG representing the conditional independence relations in a hidden Markov decision tree.

Moreover, the decision variables are linked by Markovian dynamics. Then each decision in the tree is dependent of the decision taken at that node in the previous time step. An exact forwardbackward algorithm can still be derived for the expectation step. Unfortunately, the maximization step cannot be analytically computed anymore.

6 Conclusion

We have shown that hidden Markov models with distributed state representations can provide a richer, and still efficient, modeling tool than classical HMMs. Indeed, even if an exact inference algorithm exists, it is intractable for large state space, but approximation algorithms can be used instead. These algorithms have been assessed on both synthetic and real datasets.

References

[1] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden markov models. *Mach. Learn.*, 29(2-3):245–273, November 1997. ISSN 0885-6125. doi: 10.1023/A:1007425814087. URL `http://dx.doi.org/10.1023/A:1007425814087`.

[2] C.L. Blake D.J. Newman and C.J. Merz. UCI repository of machine learning databases, 1998. URL `http://www.ics.uci.edu/$\sim$mlearn/MLRepository.html`.

[3] CKI Williams and Geoffrey E. Hinton. Mean field networks that learn to discriminate temporally distorted strings. *Connectionist Models: Proceedings of the 1990 Summer School*, pages 18–22, 1990.