Factorial Hidden Markov Models

Oussama Ennafii

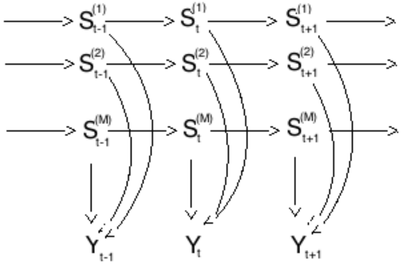
December 17, 2014

Ecole Normale Supérieure, Cachan

Introduction

- Hidden Markov Models (HMM) are widely used as learning models for time series data (such an application is speech recognition modeling).
- A generalisation of this model is what we call factorial(distributed HMM).
- In this framework, the hidden state variable is actually a vector of multiple state variables.
- Basicaly, we can represent with this model, quite easily, K^M different states with just M variables.
- A practical example where it comes handy to use this model, is when the sound recorded comes from multiple sources and we want to recognize separatly the two sounds.

Figure: The FHMM represented as a DAG.



The factorial model is represented by:

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

with:

$$P(S_t|S_{t-1}) = \prod_{m=1,\ldots,M} P(S_t^{(m)}|S_{t-1}^{(m)}) \quad (2)$$

and:

$$P(Y_t|S_t) = \mathcal{N}(Y_t, \sum_{m=1\ldots M} W^{(m)} S_t^{(m)}, C) \quad (3)$$

Inference:
The M step:

The parameters are chosen in this step to be:

$$W^{new} = (\sum_{t=1..T} Y_t < S_t^* >)(\sum_{t=1..T} < S_t S_t^* >)^\dagger \quad (4)$$

$$\pi^{(m)\ new} =< S_t^{(m)} > \quad (5)$$

$$P_{i,j}^{(m)\ new} = \frac{\sum_{t=2..T} < S_{t,i}^{(m)} S_{t,j}^{(m)} >}{\sum_{t=2..T} < S_{t,j}^{(m)} >} \quad (6)$$

$$C^{new} = \frac{1}{T}(\sum_{t=1..T} Y_t Y_t^* - \sum_{t=1..T} \sum_{t=1..M} W^{(m)} < S_t^{(m)} > Y_t^*) \quad (7)$$

Inference:

The exact E step

We define:

$$\alpha_t(S_t) = P(S_t, \{Y_\tau\}_1^t | \psi)$$

$$\beta_t(S_t) = P(\{Y_\tau\}_{t+1}^T | S_t, \psi)$$

We get through Forward-Backward algorithm:

$$P(S_t | \{Y_\tau\}_1^T, \psi) = \frac{\alpha_t(S_t)\beta_t(S_t)}{\sum_{S_t} \alpha_t(S_t)\beta_t(S_t)}$$

We deduce the means from this probability distribution.This methods complexity is:

$$O(TMK^{M+1})$$

Inference:

Inexact E step

We can use Gibbs sampling.Using the fact that a node is idependent from all other nodes, conditionally on its Markov Blanket; we sample $S_t^{(m)}$ with:

$$P(S_t^{(m)}|S_{t-1}^{(m)})P(S_{t+1}^{(m)}|S_t^{(m)})P(Y_t|S_t)$$

We can otherwise use variational techniques;In the exact EM algorithm, by choosing the $Q(S_t)$ distribution to be equal to $P(S_t|Y_t)$,we minimize the Kullback-Leiber divergence between Q and P. So we can try to impose conditions on this minimisation such that we can compute easily the posterior probabilities.

Intuitively, there are two choices:

- Completely factorised distribution:

$$Q(S_t) = \prod_{t=1..T} \prod_{m=1..M} \prod_{S_t=1..K} (\theta_{t,k}^{(m)})^{S_t^{(m)}}$$

  This yields a fixed point set of equations:

$$\theta_t^{(m)} = f(\theta_{t-1}^{(m)}, \theta_{t+1}^{(m)}, \psi)$$

- Structured distribution:

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

  with:

$$Q(S_t^{(m)}|S_{t-1}^{(m)}, h) = \prod_{k=1..K} (h_{t,k}^{(m)} \prod_{j=1..K} (P_{k,j}^{(m)})^{S_{t-1}^{(m)}})$$

Numerical results:

We generate artificially unidimensional data using a Factorial HMM with random parameters and setting K=2 and M=3. We repeat the process 20 times, we get this table reprensenting the negative log likelihood (in bits) divided by T:

Algorithm	Training Set	Test set
Naive	1.43 ± 0.23	2.61 ± 0.27
Exact	1.05 ± 0.43	1.32 ± 0.51
Variational	1.16 ± 0.85	1.41 ± 0.79