Hidden Markov Models

Thierry Artières

12 octobre 2018

Context

- Generative HMMs for sequences (Late 80's)
- Discriminative criterion for HMMs (90's)
- Structured output prediction (2000's)
- Large margin learning for HMMs (mid 2000's)
- Conditional models (CRFs) for signal labeling (mid 2000's)

Notations

variable name	value	meaning
o	$\in \mathcal{O} \equiv R^d$	single observation
o	sequence of x	observation sequence
T	integer	Length of **o**
o_t	$\in \mathcal{O} \equiv R^d$	t^{th} observation in **o**
$o_{t_1:t_2}$		subsequence $(o_{t_1}, ..., o_{t_2})$ of **o**
y	nominal $\in \mathcal{Y} \equiv 1...K$	label $\equiv$ class
y		label sequence
s	nominal $\in 1...P$	state in a markovian model
s		state sequence
q	nominal $\in 1...N$	hidden state
q		state sequence
N	integer	number of states
L	integer	number of classes / labels
λ	$R^{\lvert\lambda\rvert}$	set of parameters

Markov chain

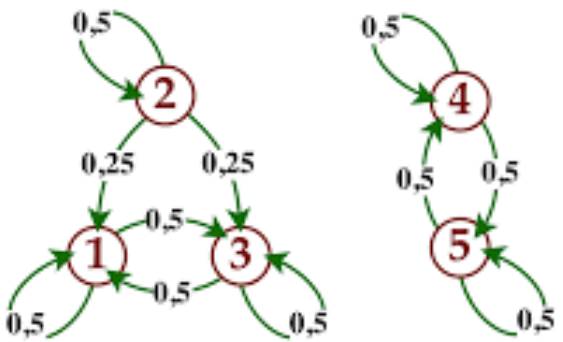
Markov property

- Modeling discrete state dynamics
- Markov property : The state (observation) at time t is independent of state at time $t' < t-1$ conditionally to state at time $t-1$
- Notations
  - Probability of initial state $\Pi_i = P(s_1 = q_i)$
  - Transition probability $a_{i,j} = P(s_t = q_j | s_{t-1} = q_i)$
- Based on the Markov property :

$$P(s_1, s_2, ...s_T) = \pi_{s_1} \prod_{i=2}^{T} a_{s_{t-1}, s_t}$$

Graphical representation

$$P = \begin{pmatrix} 1/2 & 0 & 1/2 & 0 & 0 \\ 1/4 & 1/2 & 1/4 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 1/2 \end{pmatrix}$$



Hidden Markov Model

Markov property

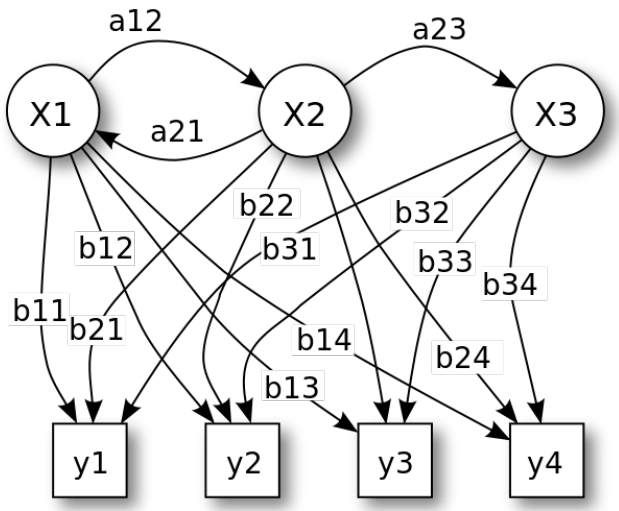
- More powerful model of sequences
  - Intermediate representation : states ($\neq$ from observations)
- A HMM is a mix of
  - A Markov chain that rules the state dynamics
  - Emission probability (on observation space) associated to states $b_i(v) = P(o = v|s_t = q_i)$
- Assumption : Markov property for state dynamics, independence assumption of observations given states

$$P(o_1, .., o_T|s_1, s_2, ...s_T) = \prod_{i=1}^{T} b_{s_t}(o_t)$$

$$P(o_1, .., o_T, s_1, s_2, ...s_T) = \pi_{s_1} b_{s_1}(o_1) \prod_{i=2}^{T} a_{s_{t-1}, s_t} b_{s_t}(o_t)$$

Hidden Markov Model

Graphical representation



The hidden feature of HMM

- Usually the state sequence is not known
- The probability of a sequence may be computed by summing over state sequences

$$p(o_1, ...o_T) = \sum_{s_1,\ldots,s_T} p(o_1, ..., o_T, s_1, ..., s_T)$$

- Naive computation : exponential cost in the length of the sequence, $O(N^T)$
- Efficient computation based on HMM underlying assumptions

Forward algorithm

Goal : compute $p(o_1, ...o_T|\lambda)$

- Idea : iteratively (on t) compute $\alpha_t(i) \stackrel{\text{def}}{=} p(o_1, ...o_t, s_t = q_i|\lambda)$
- Initialize :

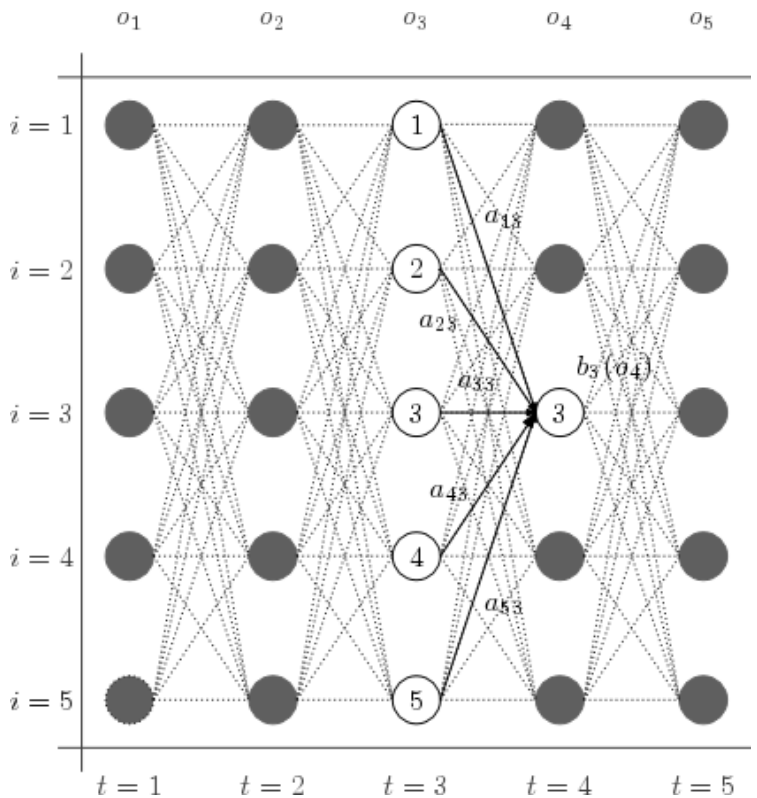
$$\alpha_1(i) \stackrel{\text{def}}{=} p(o_1, s_1 = q_i|\lambda) = \pi_i \times b_i(o_1)$$

$$\forall t \in [2, T], \alpha_t(i) = \sum_{j=1}^{N} \alpha_{t-1}(j) \times a_{s_{t-1},j} \times b_i(o_1)$$

- Complexity : $O(T \times N)$
- At the end

$$p(o_1, ...o_T|\lambda) = \sum_{j=1}^{N} \alpha_T(j)$$

Forward algorithm



Decoding : Viterbi algorithm

Goal : find best state sequence explaining an observation sequence

$$\hat{s_1}, ..., \hat{s_T} = argmax_{s_1,...,s_T} p(o_1, ..., o_T, s_1, ..., s_T)$$

Forward pass

- Forward pass : iteratively (on t) compute
  $\delta_t(i) \stackrel{\text{def}}{=} \max_{s_1,...,s_{t-1}} p(o_1, ...o_t, s_1, ..., s_{t-1}, s_t = q_i|\lambda)$

$$\delta_1(i) \stackrel{\text{def}}{=} p(o_1, s_1 = q_i|\lambda) = \pi_i \times b_i(o_1)$$

$$\forall t \in [2, T], \delta_t(i) \stackrel{\text{def}}{=} \max_{j=1}^{N} \delta_{t-1}(j) \times a_{s_{t-1},j} \times b_i(o_1)$$

- Along forward pass, memorize the best paths :

$$\Psi_1(i) \stackrel{\text{def}}{=} \text{NULL}$$

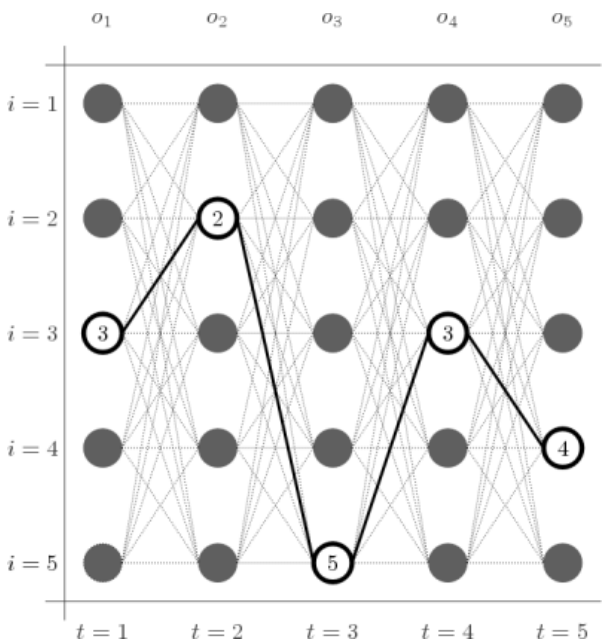
$$\forall t \in [2, T], \Psi_t(i) \stackrel{\text{def}}{=} argmax_{j=1}^{N} \delta_{t-1}(j) \times a_{s_{t-1},j} \times b_i(o_1)$$

Decoding : Viterbi algorithm

Backward pass

- Backward pass

$$\hat{s}_T = argmax_{j=1..N} \delta_T(j)$$
$$\forall t \in [N-1, 1], \hat{s}_t = \Psi_{t+1}(\hat{s})_{t+1}$$



Learning : supervised case

Countings

- Inputs : Set of observation sequences labeled with the corresponding hidden state

$$\pi_i = \frac{\text{Number of times a sequence starts in state } q_i}{\text{Number of sequences}}$$

$$a_{i,j} = \frac{\text{Number of transitions from } q_i \text{ to } q_j}{\text{Number of transitions from } q_i}$$

$$b_i(v) = \frac{\text{Number of times } v \text{is produced in state } q_i}{\text{Number of times in state } q_i}$$

Learning : unsupervised case

Instance of EM algorithm

- Inputs : Set of observation sequences
- Iterate
  - Infer (distribution on) hidden state sequence for every training sequence
  - Optimize parameters based on the joint knowledge of training sequence and inferred state sequences

Reestimation formulae

Required quantities

- Defining backward variables β in a similar way as forward variables α (similar algorithm)

$$\beta_t(i) = p(o_{t+1}, ..., o_T | s_t = q_i, \lambda)$$

- Computing necessary quantities

$$\gamma_t(i) \stackrel{\text{def}}{=} p(s_t = i | o1, ..., o_t, \lambda)$$

$$\gamma_t(i) = \frac{p(s_t = q_i, o_1, ...o_T | \lambda)}{p(o_1, ..., o_T | \lambda)}$$

$$\gamma_t(i) = \frac{\alpha_t(i) \times \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j) \times \beta_t(j)}$$

$$\xi_t(i,j) \stackrel{\text{def}}{=} p(s_t = i, s_{t+1} = j | o1, ..., o_t, \lambda)$$

$$\xi_t(i) = \frac{\alpha_t(i) \times a_{i,j} \times b_i(o_t) \times \beta_{t+1}(j)}{\sum_{j=1}^{N} \alpha_t(j) \times \beta_t(j)}$$

Reestimation formulae

- As if using countings but weighted with variables such as $\gamma_t(i)$
- For instance $\sum_{t=1}^{T} \gamma_i(t)$ = Expected number of transitions from state i in $\mathbf{o}$

HMMs training for classification tasks

- Training dataset
  - K training samples : $\left\{(\mathbf{o}^i, y^i), i \in [1; K]\right\}$
  - Each sample is labeled with class label, $y^i \in [1; L]$

- Maximum Likelihood Estimation

$$\lambda^* = argmax_\lambda \sum_{i=1}^{N} log\left(p(\mathbf{o}^i, \mathbf{y}^i|\lambda)\right)$$

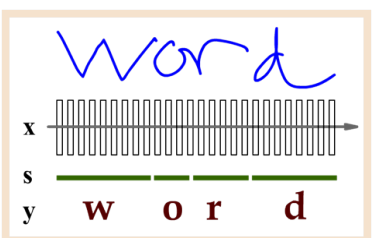
Using HMMs for labeling tasks

Principle

- One model (e.g. left-right) per label
- Build a big model by connecting all label models (transition from end states to start states)
- Given an observation sequence $\mathbf{x}$:
  - Infer the most lileky state sequence in the big model :

$$\mathbf{s}^* = argmax_{\mathbf{s}} p(\mathbf{x}, \mathbf{s})$$

$\Rightarrow$ One label sequence per Hidden state sequence
$\Rightarrow$ Multiple Hidden state sequence per label sequence



HMMs training for labelling tasks

- Training dataset
  - N training samples : $\left\{(\mathbf{o}^i, \mathbf{y}^i), i \in [1; K]\right\}$
  - Each sample is partially labeled :
    $\mathbf{o}^i$ of length T^i, $\mathbf{y}^i$ of length $M^i \leq T^i$, $\mathbf{y}^i \in [1, L]^{M^i}$

- Maximum Likelihood Estimation

$$\lambda^* = argmax_\lambda \sum_{i=1}^{N} log\left(p(\mathbf{o}^i, \mathbf{y}^i|\lambda)\right)$$

- Requires modeling $\mathbf{o}$

Gaussian HMMs

$\lambda = \{\mathbf{\Pi}, \mathbf{A}, \mu, \mathbf{\Sigma}\}$
Using standard notations

- $\mathbf{\Pi}$ stands for the initial state probabilities,
- $\mathbf{A}$ for transition probabilities,
- μ for all mean vectors $\mu = \{\mu_{s,m} | m \in [1, M], s \in [1, N]\}$,
- $\mathbf{\Sigma}$ for the set of all covariance matrices,
  $\mathbf{\Sigma} = \{\Sigma_{s,m} | m \in [1, M], s \in [1, N]\}$.

Gaussian mixture HMMs

Pdf in state s :

$$p(o|s) = \sum_{\mathbf{m}=1}^{M} P_{s,m}\mathcal{N}_{s,m}(o)$$

where

- $P_{s,m} \equiv$prior probability of the m^{th} mixture
- $\mathcal{N}_{s,m} \equiv$ Gaussian distribution
- with :

$$\mathcal{N}_{s,m}(o) = \frac{1}{\Pi^{\frac{d}{2}}|\Sigma_{s,m}|^{\frac{1}{2}}} \exp -\frac{1}{2}(o-\mu_{s,m})^{\top}\Sigma_{s,m}^{-1}(o-\mu_{s,m})$$

Gaussian mixture HMMs

Two sets of hidden variables

- State sequence (segmentation) : $\mathbf{s} = (s_1, ..., s_T)$
- Gaussian number sequence : $\mathbf{m} = (m_1, ..., m_T)$

Joint probability :

$$p(\mathbf{x}, \mathbf{y}|\lambda) = \sum_{\mathbf{s} \in S(\mathbf{y})} \sum_{\mathbf{m}} p(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}|\lambda)$$

with :

$$p(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}|\lambda) = \pi_{s_1} p_{s_1,m_1}(x_1) \prod_{t=2}^{T} a_{s_{t-1},s_t} \underbrace{p_{s_t,m_t}(x_t)}_{P_{s_t,m_t} \times \mathcal{N}_{s_t,m_t}(x_t)}$$