Hidden Markov Models

DMKM - Universitat Politècnica de Catalunya

- **Generative models:**
  - Bayes rule $\Rightarrow$ independence assumptions.
  - Able to *generate* data.
- **Conditional models:**
  - No independence assumptions.
  - Unable to generate data.

Most algorithms of both kinds make assumptions about the nature of the data-generating process, predefining a fixed model structure and only acquiring from data the distributional information.

- **Generative models:**
  - Graphical: HMM (Rabiner 1990), IOHMM (Bengio 1996). Automata-learning algorithms: *No assumptions about model structure*. VLMM (Rissanen 1983), Suffix Trees (Galil & Giancarlo 1988), CSSR (Shalizi & Shalizi 2004).
  - Non-graphical: Stochastic Grammars (Lary & Young 1990)
- **Conditional models:**
  - Graphical: discriminative MM (Bottou 1991), MEMM (McCallum et al. 2000), CRF (Lafferty et al. 2001).
  - Non-graphical: Maximum Entropy Models (Berger et al 1996).

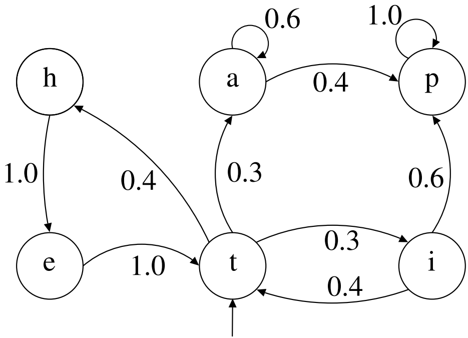
Introduction

Markov Models and Hidden Markov Models

HMM Fundamental Questions

References

- $X = (X_1, \ldots, X_T)$ sequence of random variables taking values in $S = \{s_1, \ldots, s_N\}$
- Markov Properties
  - Limited Horizon:
    $P(X_{t+1} = s_k \mid X_1, \ldots, X_t) = P(X_{t+1} = s_k \mid X_t)$
  - Time Invariant (Stationary):
    $P(X_{t+1} = s_k \mid X_t) = P(X_2 = s_k \mid X_1)$
- Transition matrix:
  $a_{ij} = P(X_{t+1} = s_j \mid X_t = s_i)$; $\quad a_{ij} \geq 0,\ \forall i, j;\ \sum_{j=1}^{N} a_{ij} = 1,\ \forall i$
- Initial probabilities (or extra state s_0):
  $\pi_i = P(X_1 = s_i)$; $\quad \sum_{i=1}^{N} \pi_i = 1$

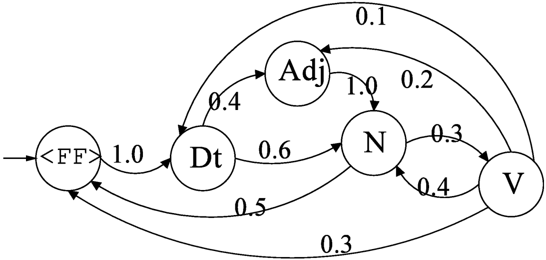


Sequence probability:

$$
\begin{aligned}
P(X_1, .., X_T) &= \\
&= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_1X_2)\ldots P(X_T \mid X_1..X_{T-1}) \\
&= P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_2)\ldots P(X_T \mid X_{T-1}) \\
&= \pi_{X_1} \prod_{t=1}^{T-1} a_{X_tX_{t+1}}
\end{aligned}
$$

- States and Observations
- Emission Probability:
  $$b_{ik} = P(O_t = k \mid X_t = s_i)$$
- Used when underlying events probabilistically generate surface events:
  - PoS tagging (hidden states: PoS tags, observations: words)
  - ASR (hidden states: phonemes, observations: sound)
  - ...
- Trainable with unannotated data. Expectation Maximization (EM) algorithm.
- arc-emission *vs* state-emission

Example: PoS Tagging



Emission probabilities	.	the	this	cat	kid	eats	runs	fish	fresh	little	big
<FF>	1.0										
Dt		0.6	0.4								
N				0.6	0.1			0.3			
V						0.7	0.3				
Adj									0.3	0.3	0.4

1. **Observation probability (decoding):** Given a model $\mu = (A, B, \pi)$, how we do efficiently compute how likely is a certain observation ? That is, $P_{\mu}(O)$
2. **Classification:** Given an observed sequence O and a model μ, how do we choose the state sequence $(X_1, \ldots, X_T)$ that best explains the observations?
3. **Parameter estimation:** Given an observed sequence O and a space of possible models, each with different parameters (A, B, π), how do we find the model that best explains the observed data?

- Let $O = (o_1, \ldots, o_T)$ observation sequence.
- For any state sequence $X = (X_1, \ldots, X_T)$, we have:

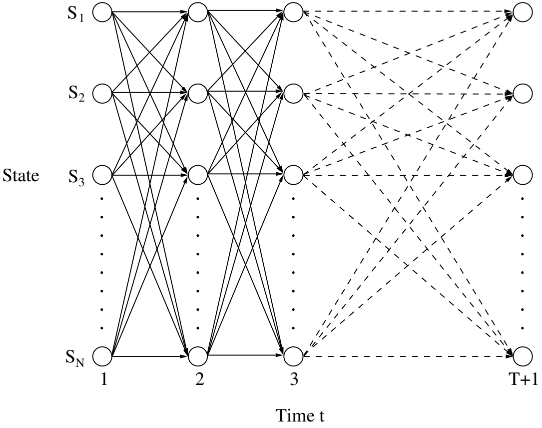
$$P_\mu(O \mid X) = \prod_{t=1}^{T} P_\mu(o_t \mid X_t) = b_{X_1 o_1} b_{X_2 o_2} \ldots b_{X_T o_T}$$

- $P_\mu(X) = \pi_{X_1} a_{X_1 X_2} a_{X_2 X_3} \ldots a_{X_{T-1} X_T}$
- $P_\mu(O) = \sum_X P_\mu(O, X) = \sum_X P_\mu(O \mid X) P_\mu(X)$

$$= \sum_{X_1 \ldots X_T} \pi_{X_1} b_{X_1 o_1} \prod_{t=2}^{T} a_{X_{t-1} X_t} b_{X_t o_t}$$

- Complexity: $\mathcal{O}(TN^T)$
- Dynammic Programming: Trellis/lattice. $\mathcal{O}(TN^2)$

Trellis



Fully connected HMM where one can move from any state to any other at each step. A node $\{s_i, t\}$ of the trellis stores information about state sequences which include $X_t = i$.

Forward & Backward computation

Forward procedure $\mathcal{O}(TN^2)$

We store $\alpha_i(t)$ at each trellis node $\{s_i, t\}$.

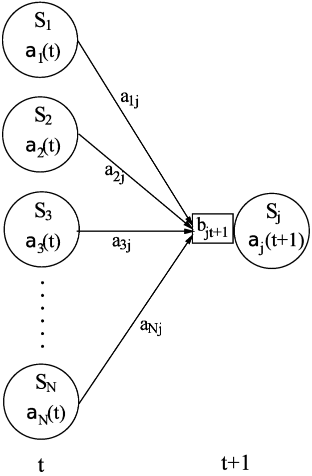
$\alpha_i(t) = P_\mu(o_1 \dots o_t, X_t = i)$ Probability of emmiting $o_1 \dots o_t$ and reach state s_i at time t.

1. Inicialization: $\alpha_i(1) = \pi_i b_{io_1}; \quad \forall i = 1 \dots N$
2. Induction: $\forall t : 1 \leq t < T$

$$\alpha_j(t+1) = \sum_{i=1}^{N} \alpha_i(t) a_{ij} b_{jo_{t+1}}; \quad \forall j = 1 \dots N$$

3. Total: $P_\mu(O) = \sum_{i=1}^{N} \alpha_i(T)$

Forward computation



Closeup of the computation of forward probabilities at one node. The forward probability $\alpha_j(t+1)$ is calculated by summing the product of the probabilities on each incoming arc with the forward probability of the originating node.

Backward procedure $\mathcal{O}(TN^2)$

We store $\beta_i(t)$ at each trellis node $\{s_i, t\}$.

$$\beta_i(t) = P_\mu(o_{t+1} \dots o_T \mid X_t = i)$$

Probability of emmiting $o_{t+1} \dots o_T$ given we are in state s_i at time t.

1. Inicialization: $\beta_i(T) = 1 \quad \forall i = 1 \dots N$
2. Induction: $\forall t : 1 \leq t < T$

$$\beta_i(t) = \sum_{j=1}^{N} a_{ij} b_{jo_{t+1}} \beta_j(t+1) \qquad \forall i = 1 \dots N$$

3. Total: $P_\mu(O) = \sum_{i=1}^{N} \pi_i b_{io_1} \beta_i(1)$

Combination

$$P_\mu(O, X_t = i) = P_\mu(o_1 \dots o_{t-1}, X_t = i, o_t \dots o_T)$$
$$= \alpha_i(t)\beta_i(t)$$

$$P_\mu(O) = \sum_{i=1}^{N} \alpha_i(t)\beta_i(t) \qquad \forall t : 1 \leq t \leq T$$

Forward and Backward procedures are particular cases of this equation when $t = 1$ and $t = T$ respectively.

Question 2. Best state sequence

- Most likely path for a given observation O:

$$\begin{aligned}\operatorname*{argmax}_X P_\mu(X \mid O) &= \operatorname*{argmax}_X \frac{P_\mu(X,O)}{P_\mu(O)} \\ &= \operatorname*{argmax}_X P_\mu(X,O) \quad \text{(since } O \text{ is fixed)}\end{aligned}$$

- Compute the best sequence with the same recursive approach than in FB: Viterbi algorithm, $\mathcal{O}(TN^2)$.

- $\delta_j(t) = \max\limits_{X_1 \ldots X_{t-1}} P_\mu(X_1 \ldots X_{t-1} s_j, o_1 \ldots o_t)$

  Highest probability of any sequence reaching state s_j at time t after emmitting $o_1 \ldots o_t$

- $\psi_j(t) = last(\operatorname*{argmax}\limits_{X_1 \ldots X_{t-1}} P_\mu(X_1 \ldots X_{t-1} s_j, o_1 \ldots o_t))$

  Last state (X_{t-1}) in highest probability sequence reaching state s_j at time t after emmitting $o_1 \ldots o_t$

Viterbi algorithm

1. Initialization: $\forall j = 1 \ldots N$
   $\delta_j(1) = \pi_j b_{jo_1}$
   $\psi_j(1) = 0$
2. Induction: $\forall t : 1 \leq t < T$
   $\delta_j(t+1) = \max_{1 \leq i \leq N} \delta_i(t) a_{ij} b_{jo_{t+1}} \quad \forall j = 1 \ldots N$
   $\psi_j(t+1) = \operatorname{argmax}_{1 \leq i \leq N} \delta_i(t) a_{ij} \quad \forall j = 1 \ldots N$
3. Termination: backwards path readout.
   - $\hat{X}_T = \operatorname{argmax}_{1 \leq i \leq N} \delta_i(T)$
   - $\hat{X}_t = \psi_{\hat{X}_{t+1}}(t+1)$
   - $P(\hat{X}) = \max_{1 \leq i \leq N} \delta_i(T)$

Question 3. Parameter Estimation

Obtain model parameters (A, B, π) for the model μ that maximizes the probability of given observation O:

$$(A, B, \pi) = \underset{\mu}{\operatorname{argmax}}\, P_{\mu}(O)$$

Baum-Welch algorithm

- Baum-Welch algorithm (*aka* Forward-Backward):
  1. Start with an initial model μ_0 (uniform, random, MLE...)
  2. Compute observation probability (F&B computation) using current model μ.
  3. Use obtained probabilities as data to reestimate the model, computing $\hat{\mu}$
  4. Let $\mu = \hat{\mu}$ and repeat until no significant improvement.
- Iterative hill-climbing: Local maxima.
- Particular application of Expectation Maximization (EM) algorithm.
- EM Property: $P_{\hat{\mu}}(O) \geq P_{\mu}(O)$

- $\gamma_i(t) = P_\mu(X_t = i \mid O) = \dfrac{P_\mu(X_t = i, O)}{P_\mu(O)} = \dfrac{\alpha_i(t)\beta_i(t)}{\sum_{k=1}^{N} \alpha_k(t)\beta_k(t)}$

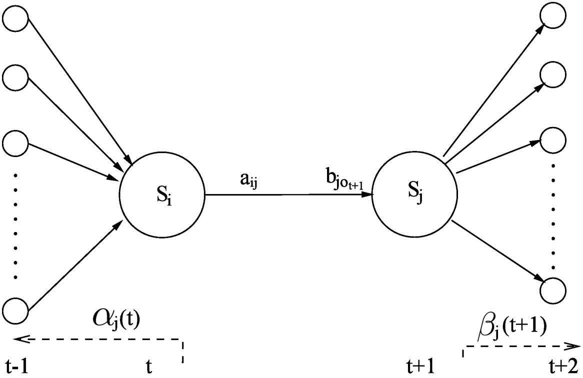
  Probability of being at state s_i at time t given observation O.

- $\varphi_t(i,j) = P_\mu(X_t = i, X_{t+1} = j \mid O) = \dfrac{P_\mu(X_t = i, X_{t+1} = j, O)}{P_\mu(O)}$

  $$= \frac{\alpha_i(t) a_{ij} b_{j o_{t+1}} \beta_j(t+1)}{\sum_{k=1}^{N} \alpha_k(t)\beta_k(t)}$$

  probability of moving from state s_i at time t to state s_j at time $t+1$, given observation sequence O. Note that $\gamma_i(t) = \sum_{j=1}^{N} \varphi_t(i,j)$

$\sum_{t=1}^{T-1} \gamma_i(t)$ Expected number of transitions from state s_i in O.	$\sum_{t=1}^{T-1} \varphi_t(i,j)$ Expected number of transitions from state s_i to s_j in O.



Given an observation O, the model μ Probability $\varphi_t(i,j)$ of moving from state s_i at time t to state s_j at time $t+1$ given observation O.

Iterative reestimation

$$\hat{\pi}_i = \text{Expected frequency in state } s_i \text{ at time } (t=1) = \gamma_i(1)$$

$$\hat{a}_{ij} = \frac{\text{Expected number of transitions from } s_i \text{ to } s_j}{\text{Expected number of transitions from } s_i} = \frac{\sum_{t=1}^{T-1} \varphi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

$$\hat{b}_{jk} = \frac{\text{Expected number of emissions of } k \text{ from } s_j}{\text{Expected number of visits to } s_j} = \frac{\sum_{\{t:\, 1 \le t \le T,\ o_t = k\}} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Introduction

Markov Models and Hidden Markov Models

HMM Fundamental Questions

References

References

- C. Manning & H. Schütze, **Foundations of Statistical Natural Language Processing**. The MIT Press. Cambridge, MA. May 1999.
- S.L. Lauritzen, **Graphical Models**. Oxford University Press, 1996
- L.R. Rabiner, **A tutorial on hidden Markov models and selected applications in speech recognition**. Proceedings of the IEEE, Vol. 77, num. 2, pg 257-286, 1989.