A tutorial on hidden markov models and selected applications in speech recognition

Author
Rabiner L.

Journal
Proceding of the IEEE,1989

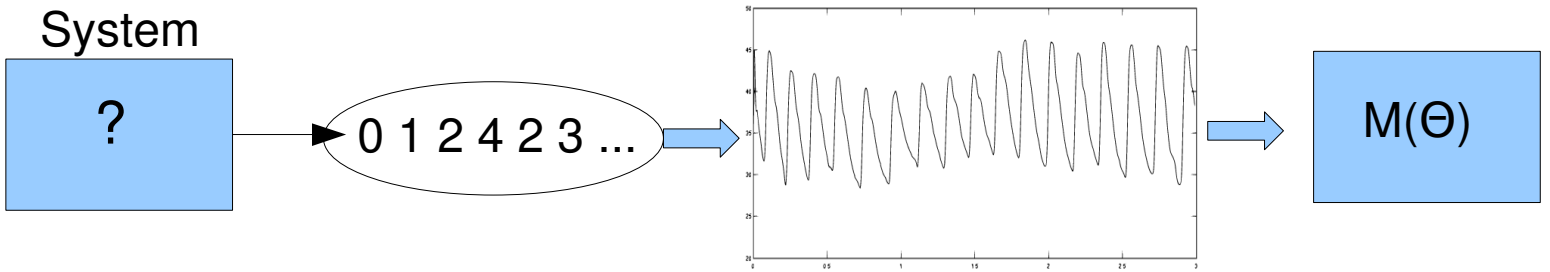
Speaker
Simone Tognetti

Outline

- Introduction
  - Graph models for signal
- Learning a model
  - ML
  - EM
- Markov Chain
- HMM
  - Solve HMMs
- Extension of HMMs
- Applications
- Conclusion

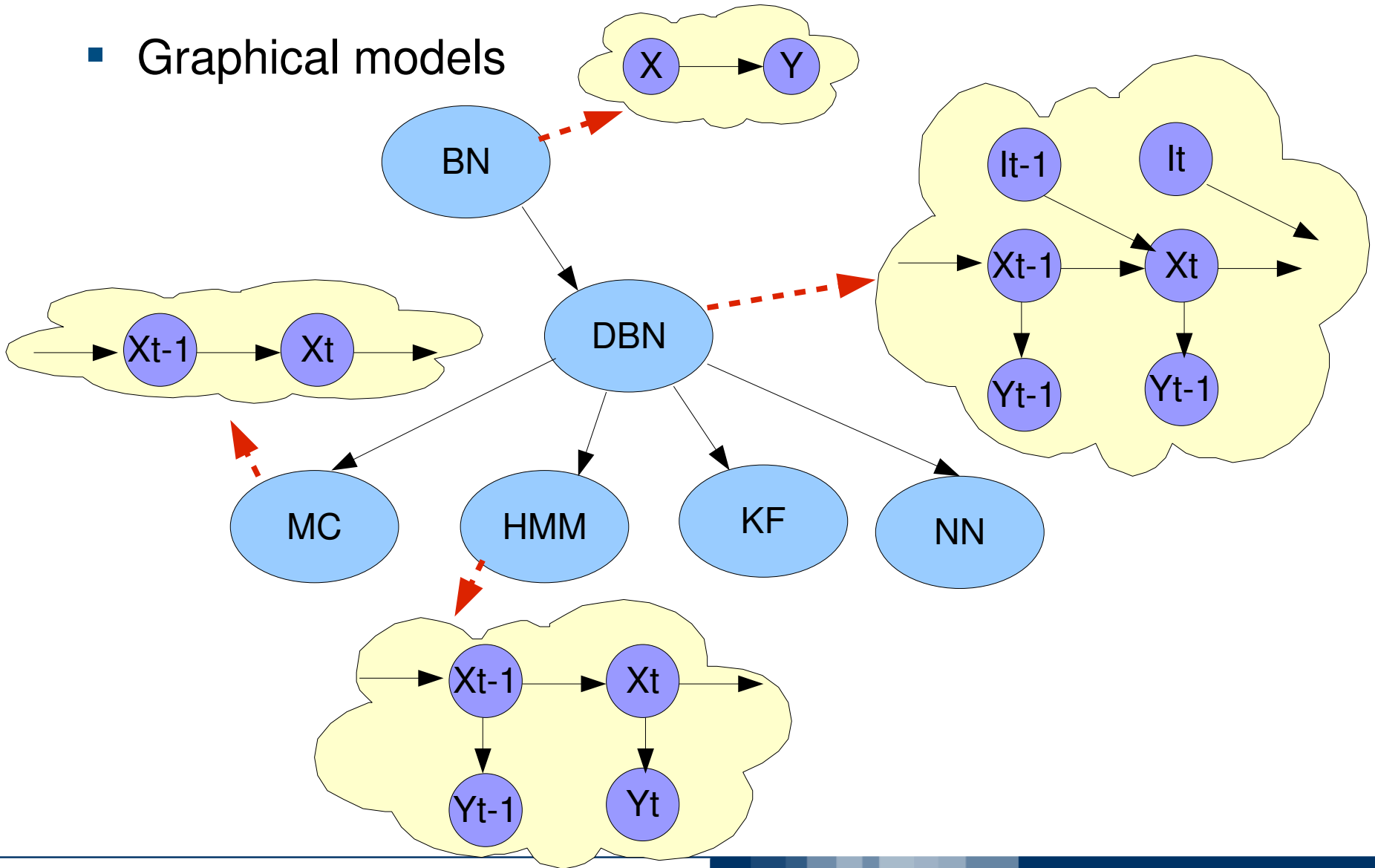
Introduction: we can model what we see

- Real word system generate observable outputs
- Outputs can be viewed as stochastic signals
- We can build a model of that signal because :
  - the model provide information concerning how to do the processing of the signal (ex. how to remove noise)
  - the model give informations about the system that have generated the signal
  - from the model we can make predictions.



Which model we choose?

- Graphical models



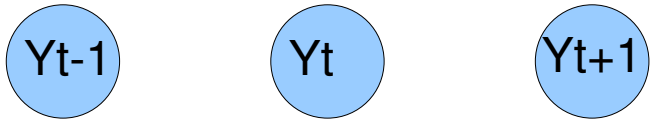
Outline

- Introduction
  - Graph models for signal
- Learning a model
  - ML
  - EM
- Markov Chain
- HMM
  - Solve HMMs
- Extension of HMMs
- Applications
- Conclusion

Learnig BN: ML

- Maximum likelihood estimation
  - Estimation of the parameters of a stochastic model when all the random variable are observable.
  - Find the parameters that maximize the likelihood

Sequence of random variable iid ~ N(μ,σ^2)



$$\mathcal{L}(\theta) = \log P(\{Yt\}|\theta) = \log \prod_{t=1}^{T} \left(\frac{1}{2\pi\sigma^2}\right) exp\left\{\frac{(Yt-\mu)^2}{2\sigma^2}\right\}$$

$$\begin{cases} \dfrac{\partial \mathcal{L}(\theta)}{\partial \mu} = \dfrac{1}{T}\displaystyle\sum_{t=1}^{T} Yt \\ \dfrac{\partial \mathcal{L}(\theta)}{\partial \sigma^2} = \dfrac{1}{T}\displaystyle\sum_{t=1}^{T} (Yt-\mu)^2 \end{cases}$$

Learning BN: EM

- Expectation Maximization (EM)

  Learn parameters of a stochastic model when some variables are hidden. (ie. HHM)
  - Learn means to find something that we don't see
  - A simple example

$$\mathcal{L}(\theta) = \log P(Y|\theta) = \log \sum_X P(Y, X|\theta) = \log \sum_X Q(X) \frac{P(Y, X|\theta)}{Q(X)}$$

$$\mathcal{L}(\theta) \geq E(Q, \theta) - H(Q) = F(Q, \theta)$$

- EM maximize the likelihood in a iterative with two steps
  - Expectation $Q_{k+1} \leftarrow arg \max_Q F(Q, \theta_k)$
  - Maximization $\theta_{k+1} \leftarrow arg \max_\theta F(Q_{k+1}, \theta)$

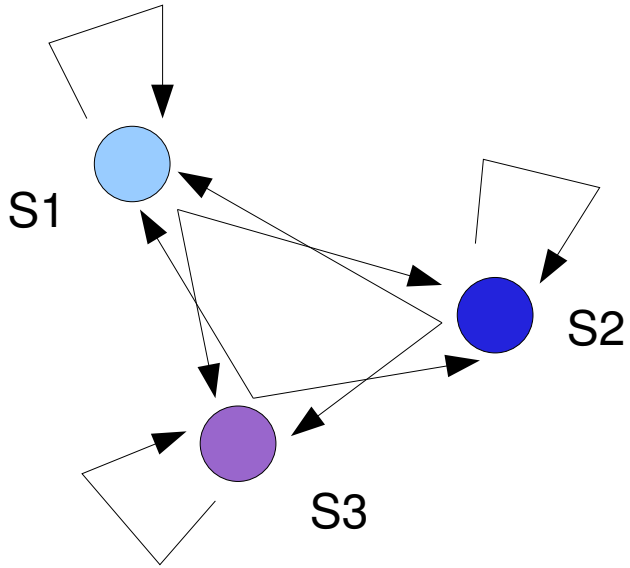
Outline

- Introduction
  - Graph models for signal
- Learning a model
  - ML
  - EM
- Markov Chain
- HMM
  - Solve HMMs
- Extension of HMMs
- Applications
- Conclusion

Discrete Markov Process

- Set of state S
- Transition Matrix M

$$M = \begin{pmatrix} P(S1|S1) & P(S2|S1) & P(S3|S1) \\ P(S1|S2) & P(S2|S2) & P(S3|S2) \\ P(S1|S3) & P(S2|S3) & P(S3|S3) \end{pmatrix}$$



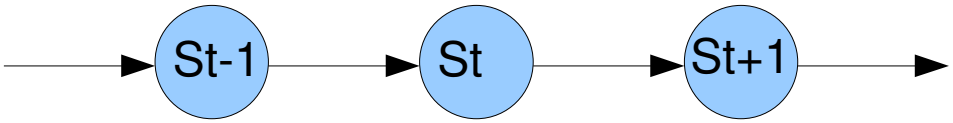
- Prior Probability

  $\pi(0) = [\ P(S1)\ P(S2)\ P(S3)\ ]$

- Marginal (Posterior probability)

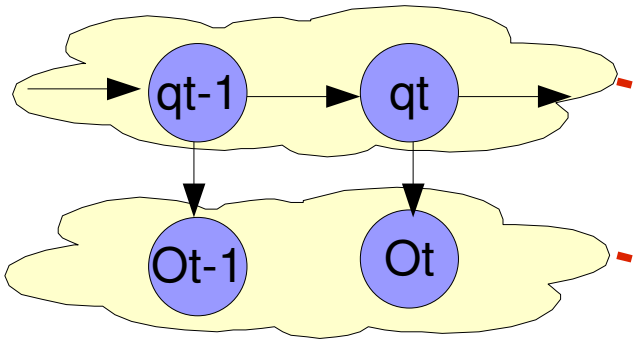
  $\pi(t) = \pi(0)\ M^{(t)}$

- DBN



Hidden Markov Models

- DBN



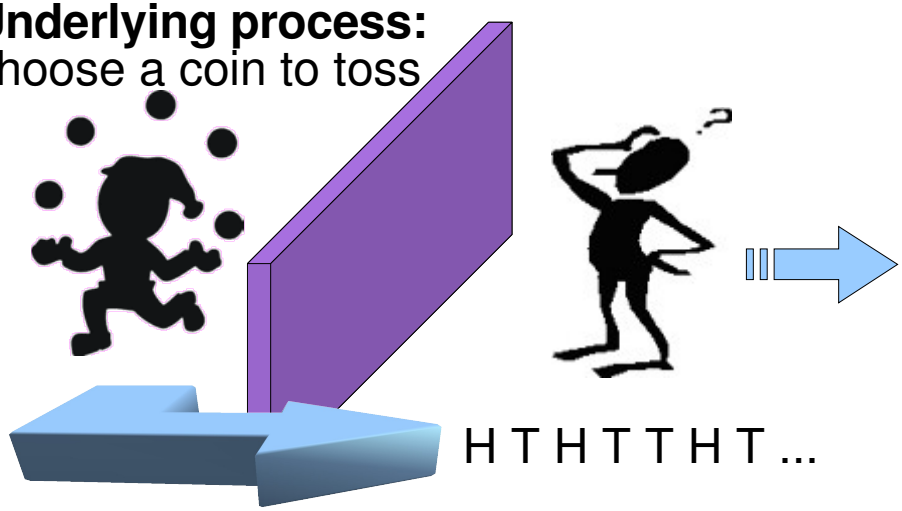
Underlying Markov Process that is not Observable

Output Markov Process that is Observable

- How to think about them?

.. i.e. coin tossing

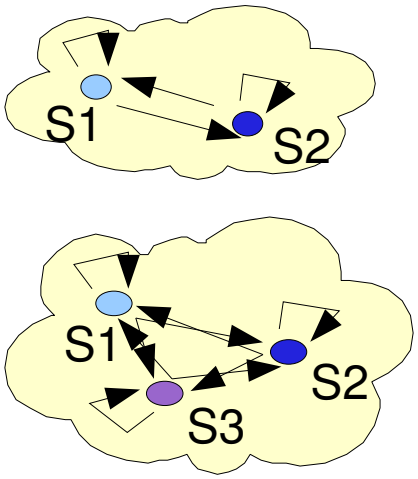
Underlying process:
choose a coin to toss



H T H T T H T ...

Output process: depends on the chosen coin

Hypothesis on the underlying model:



Hidden Markov Models

- Set of state: qt in { S1, S2, S3, ... , SN }
- Underlying process transition matrix A

  aij=P(qt=Si|qt-1=Sj)
- Output process matrix B (Produce a set of observation O)
  - Discrete: Ot in { v1,v2, .. vD}

    bi(k)= P(Ot = vk|qt = Si), B is N x D
  - Continuous: Ot in R

    bi = P(Ot|qt = St) , B is a vector of distribution
- Prior probability on states

  π(0)= [P(q0 = S1), P(q0 = S2), ... P(q0 = SN)]

Learning HMM

- Basic problems

  We have an HMM $\lambda = (A,B,\pi)$ and a sequence of observation $O = O1,O2,...On$

  1. $P(O|\lambda)$: Probability that an observation sequence came from a given HMM
  2. $P(q1,q2,...,qt|O)$: Probability to have a specific state sequence given the observation sequence
  3. How to adjust λ to maximize $P(O|\lambda)$: Learning the HMM

Problem 1: P(O|λ)

- Given a state sequence $Q = q_1\, q_2 \cdots q_T$
- Probability of the observation sequence

$$P(O|Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdots b_{q_T}(O_T).$$

- Probability of a state sequence

$$P(Q|\lambda) = \pi_{q1} a_{q_1 q_2} a_{q_2 q_3} \cdots a_{q_{T-1} q_T}.$$

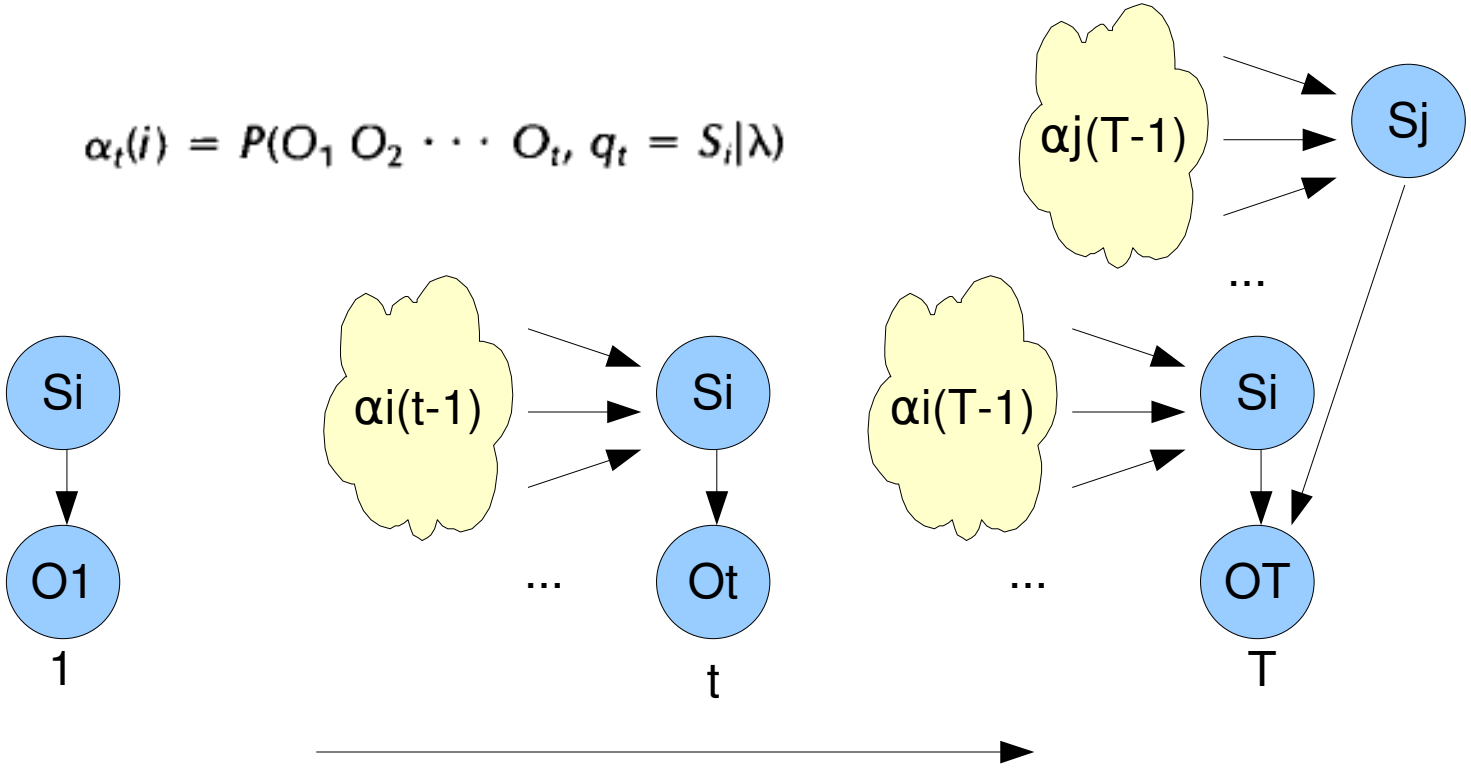
- Join probability of Q and O $P(O, Q|\lambda) = P(O|Q, \lambda)\, P(Q, \lambda).$
- Probability of O by summation over all state sequence

$$P(O|\lambda) = \sum_{\text{all } Q} P(O|Q, \lambda)\, P(Q|\lambda)$$

Problem 1: P(O|λ)

- Forward procedure: compute a forward variable

$$\alpha_t(i) = P(O_1\, O_2 \cdots O_t,\, q_t = S_i|\lambda)$$

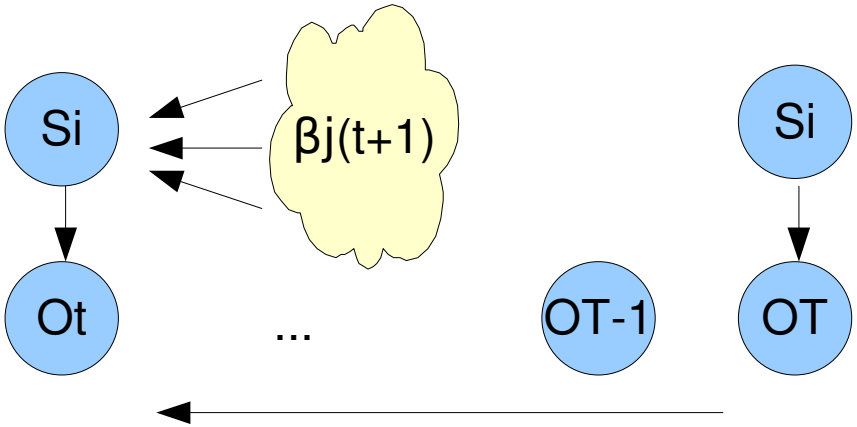


$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

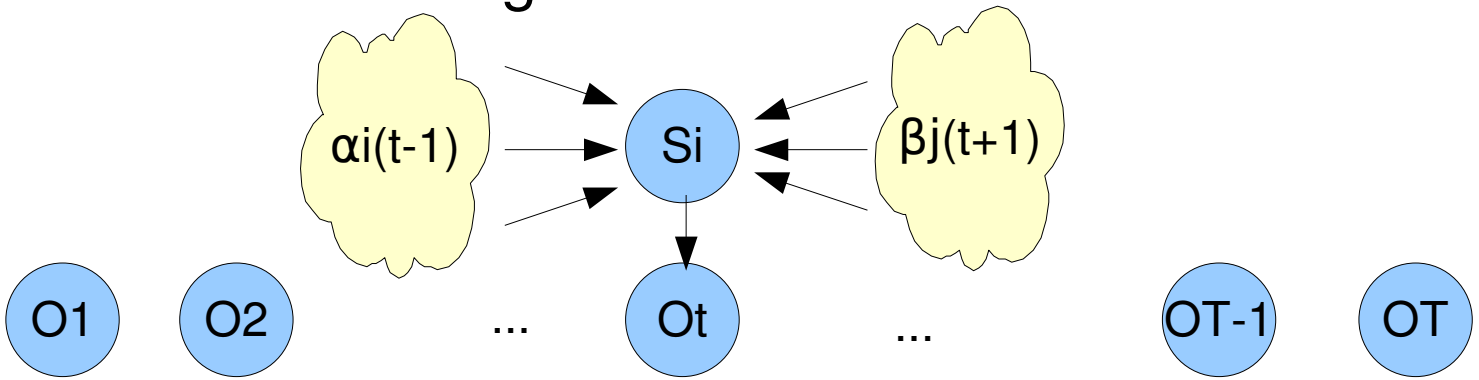
EM for HMM : Forward Backward procedure

- Forward Variable $\alpha_t(i) = P(O_1\, O_2 \cdots O_t, q_t = S_i|\lambda)$
- Backward variable $\beta_t(i) = P(O_{t+1}\, O_{t+2} \cdots O_T|q_t = S_i, \lambda)$

E-step



- Alfa and Beta together



Problem 2: P(q1,q2, ..., qt|O)

- Probability of being in state Si at time t given the observation sequence

$$\gamma_t(i) = P(q_t = S_i|O, \lambda)$$

- Expression in terms of forward and backward variable

$$\gamma_t(i) = \frac{\alpha_t(i)\,\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\,\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\,\beta_t(i)}$$

- Possible solution: find local best state

$$q_t = \underset{1 \le i \le N}{\operatorname{argmax}} [\gamma_t(i)], \qquad 1 \le t \le T.$$

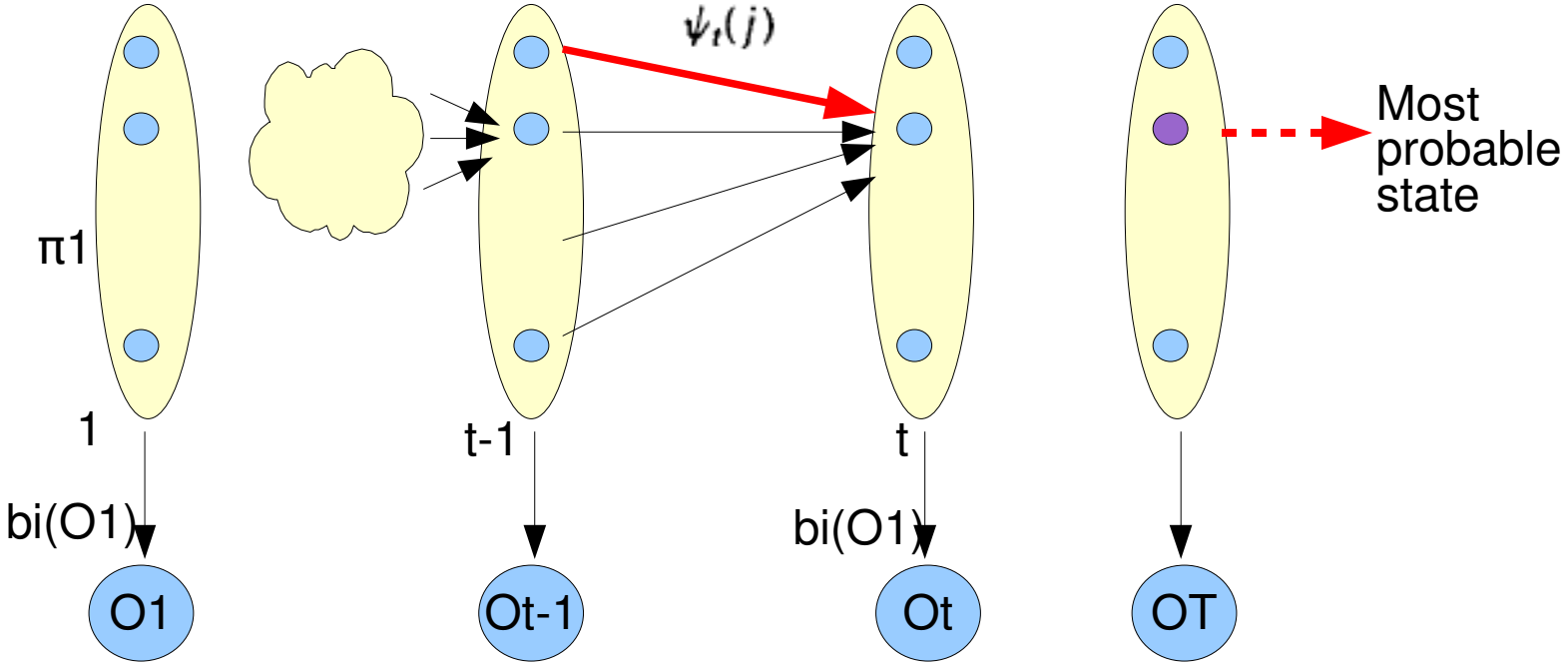
  - but possible solution are not ammissible state sequence

Problem 2: P(q1,q2, ..., qt|O)

- Viterbi algorithm
  - Best score along a single path

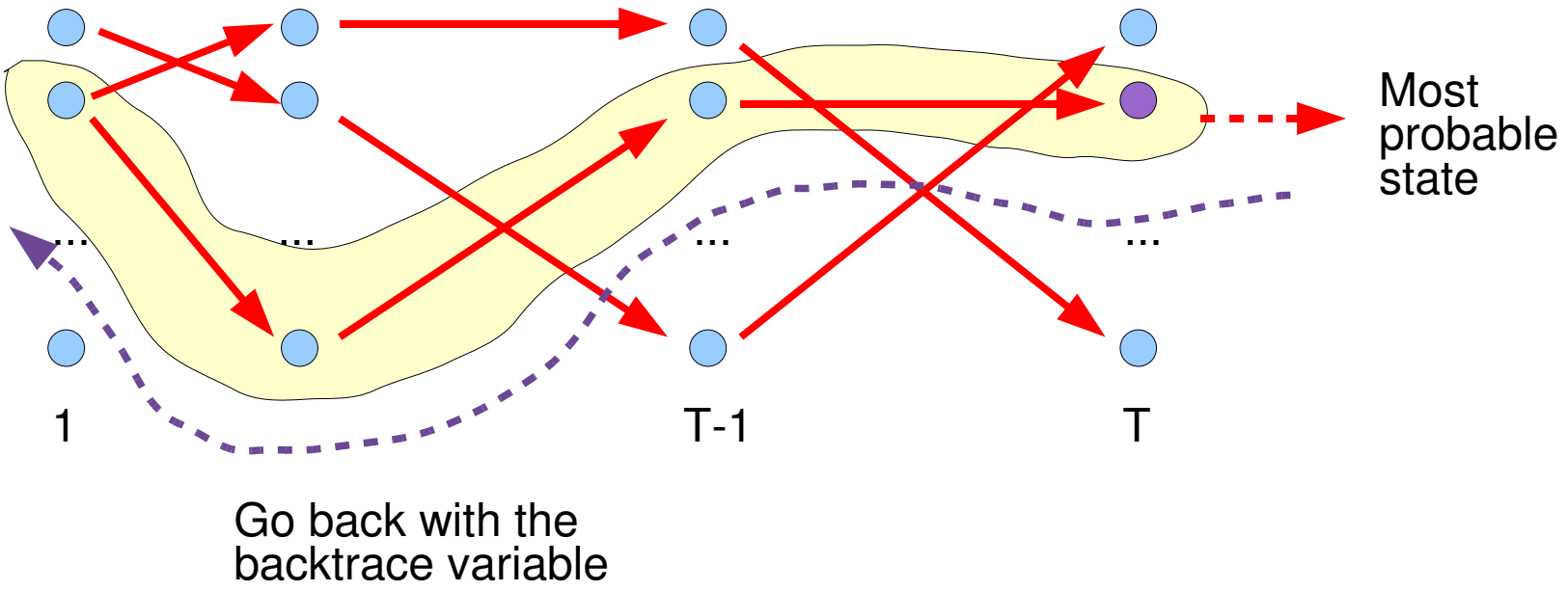
$$\delta_t(i) = \max_{q_1, q_2, \cdots, q_{t-1}} P[q_1\, q_2 \cdots q_t = i,\, O_1\, O_2 \cdots O_t | \lambda]$$

  - Keep a track of the best path that can reach the state j at time t



Problem 2: P(q1,q2, ..., qt|O)

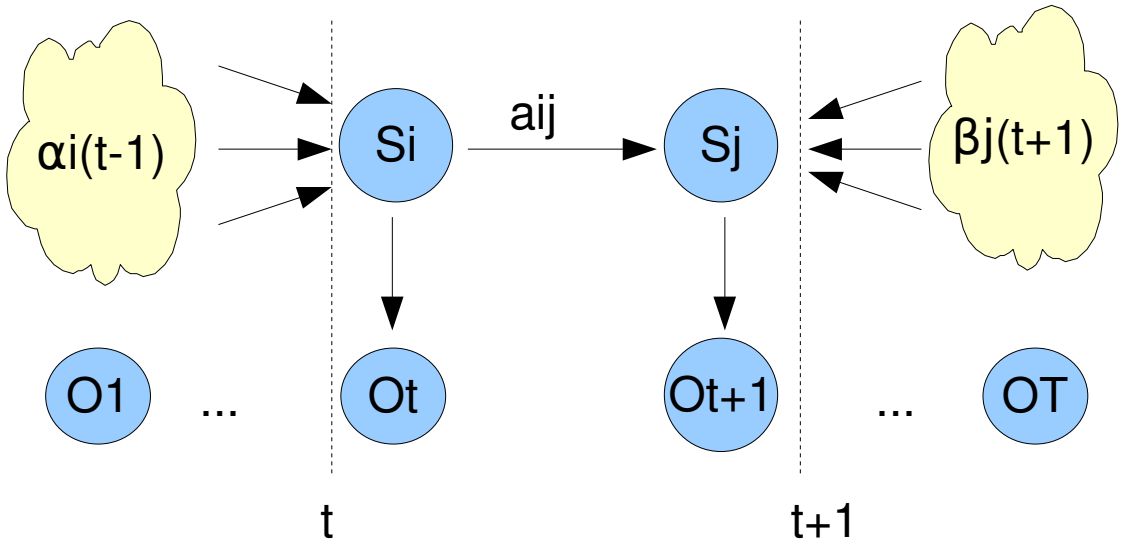
- Global behaviour



Problem 3: Learning λ

- Baum-Welch (EM algorithm for HMM)
  - Probability of being in state i at time t and in state j at time t+1

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda).$$



Extract the effect of parameter aij

Problem 3: Learning λ

- Baum-Welch (EM algorithm for HMM)
  - Expression with forward and backward variable

$$\xi_t(i, j) = \frac{\alpha_t(i)\, a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i)\, a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(O_{t+1})\, \beta_{t+1}(j)}$$

  - Relation between $\gamma_t(i)$ and $\xi_t(i, j)$ $\quad \gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j).$
  - Expected number of transition from Si $\sum_{t=1}^{T-1} \gamma_t(i)$
  - Expected number of transition from Si to Sj $\sum_{t=1}^{T-1} \xi_t(i, j)$

Problem 3: Learning λ

- Baum-Welch: estimation of parameters

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } S_i \text{ at time } (t = 1) = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } S_i \text{ to state } S_j}{\text{expected number of transitions from state } S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j} = \frac{\sum_{t=1 \text{ s.t. } O_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}.$$

Problem 3: Learning λ

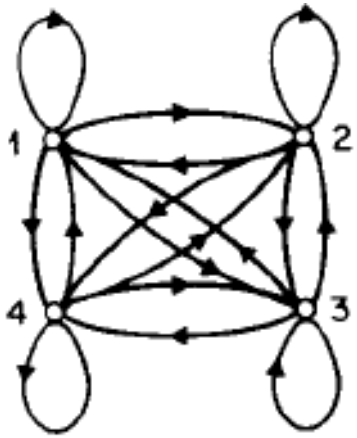
- Baum-Welch: Algorithm
  1. Initial estimation of λ
  2. E-step: Compute α, β
  3. M-Step: New estimation of parameter λ'
  4. Repeat from 2 until convergence
- EM : if we compute the likelihood we obtain same equation for the M-step
- Problem: Learning is sensible to initial parameter value when we have continuous observations

Outline

- Introduction
  - Graph models for signal
- Learning a model
  - ML
  - EM
- Markov Chain
- HMM
  - Solve HMMs
- Extension of HMMs
- Applications
- Conclusion

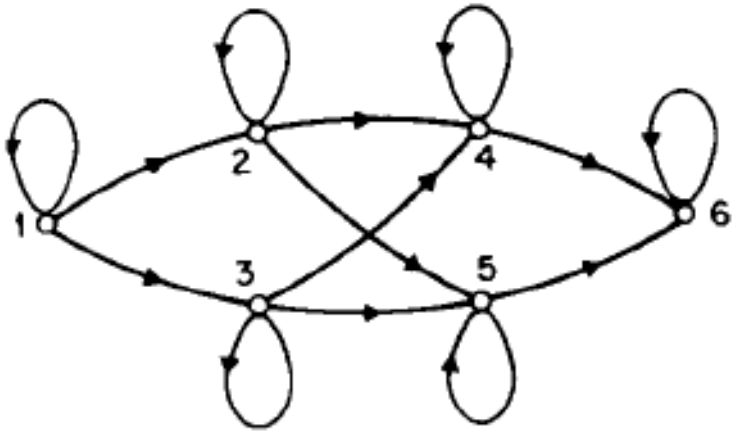
Type of HMMs

- Type change with the structure of the transition matrix of the underling process



Ergodic HMM with 4 state

Left to right HMM with 4 state



Parallel path left to right HMM with 6 state

Type of HMMs

- Continuous observations: we can model the distribution probability of the output with a mixture

$$b_j(\boldsymbol{O}) = \sum_{m=1}^{M} c_{jm} \mathfrak{N}[\boldsymbol{O}, \mu_{jm}, \boldsymbol{U}_{jm}], \quad 1 \le j \le N$$

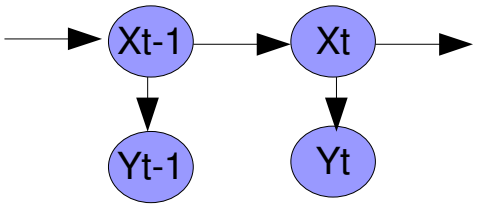
Prior vector

Mean vector

Covariance matrix

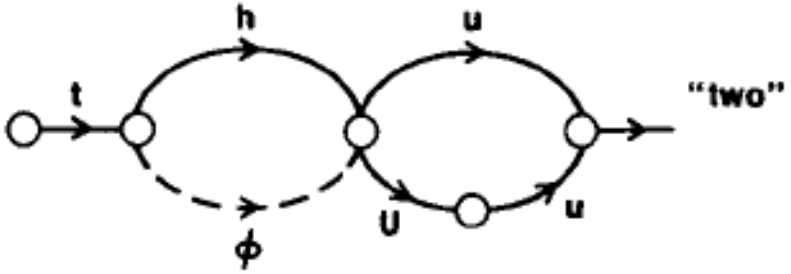
New parameter to estiamate

- Kalman filter



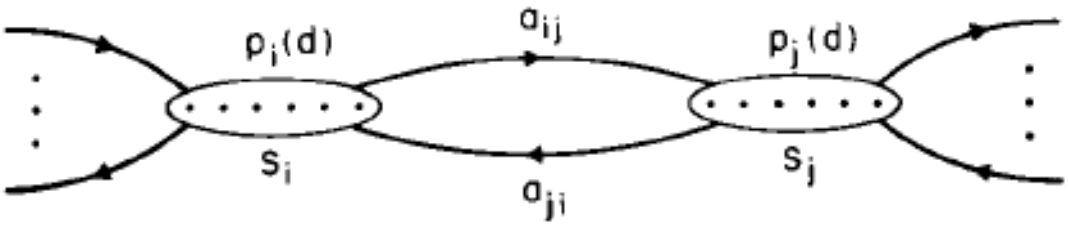
$$\begin{cases} X_t = AX_{t-1} + w_t \\ Y_t = CX_t + v_t \end{cases}$$

- Null transition



Other types and issue

- Explicit state duration



- Distance within two HMM $\lambda_1 = (A_1, B_1, \pi_1)$ $\lambda_2 = (A_2, B_2, \pi_2)$

$$D(\lambda_1, \lambda_2) = \frac{1}{T}[\log P(O^{(2)}|\lambda_1) - \log P(O^{(2)}|\lambda_2)]$$

$$D_s(\lambda_1, \lambda_2) = \frac{D(\lambda_1\lambda_2) + D(\lambda_2, \lambda_1)}{2}.$$ Symmetric version

- Scaling

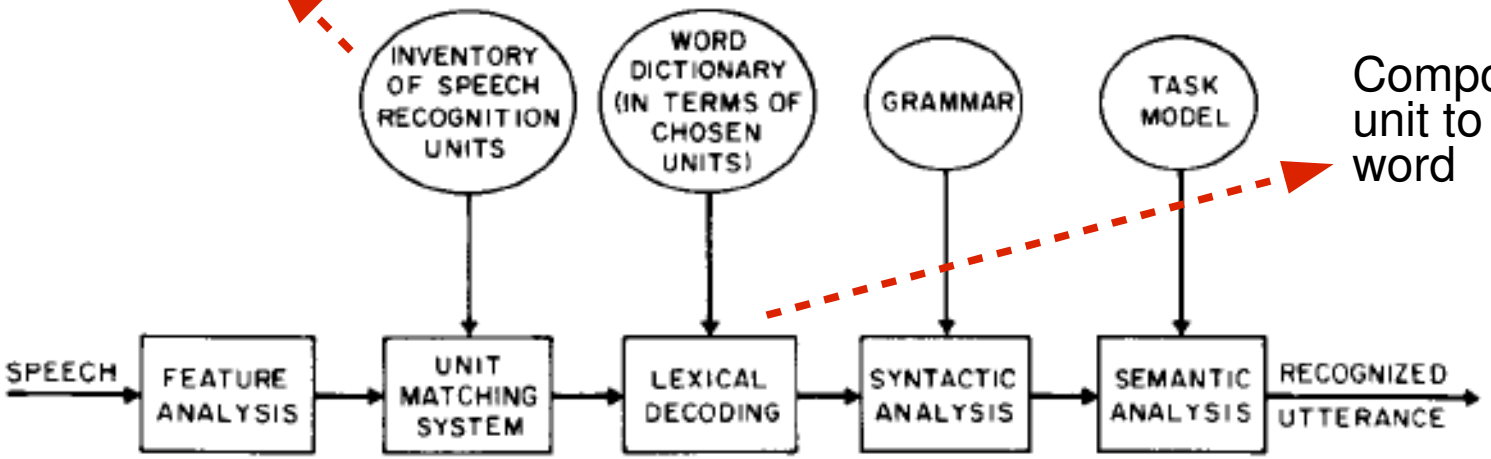
Outline

- Introduction
  - Graph models for signal
- Learning a model
  - ML
  - EM
- Markov Chain
- HMM
  - Solve HMMs
- Extension of HMMs
- Applications
- Conclusion

Speech recognition

- A generic framework

Mapping from HMM to a voice unit.
Given the observation choose the
HMM that give the best P(O|λ)



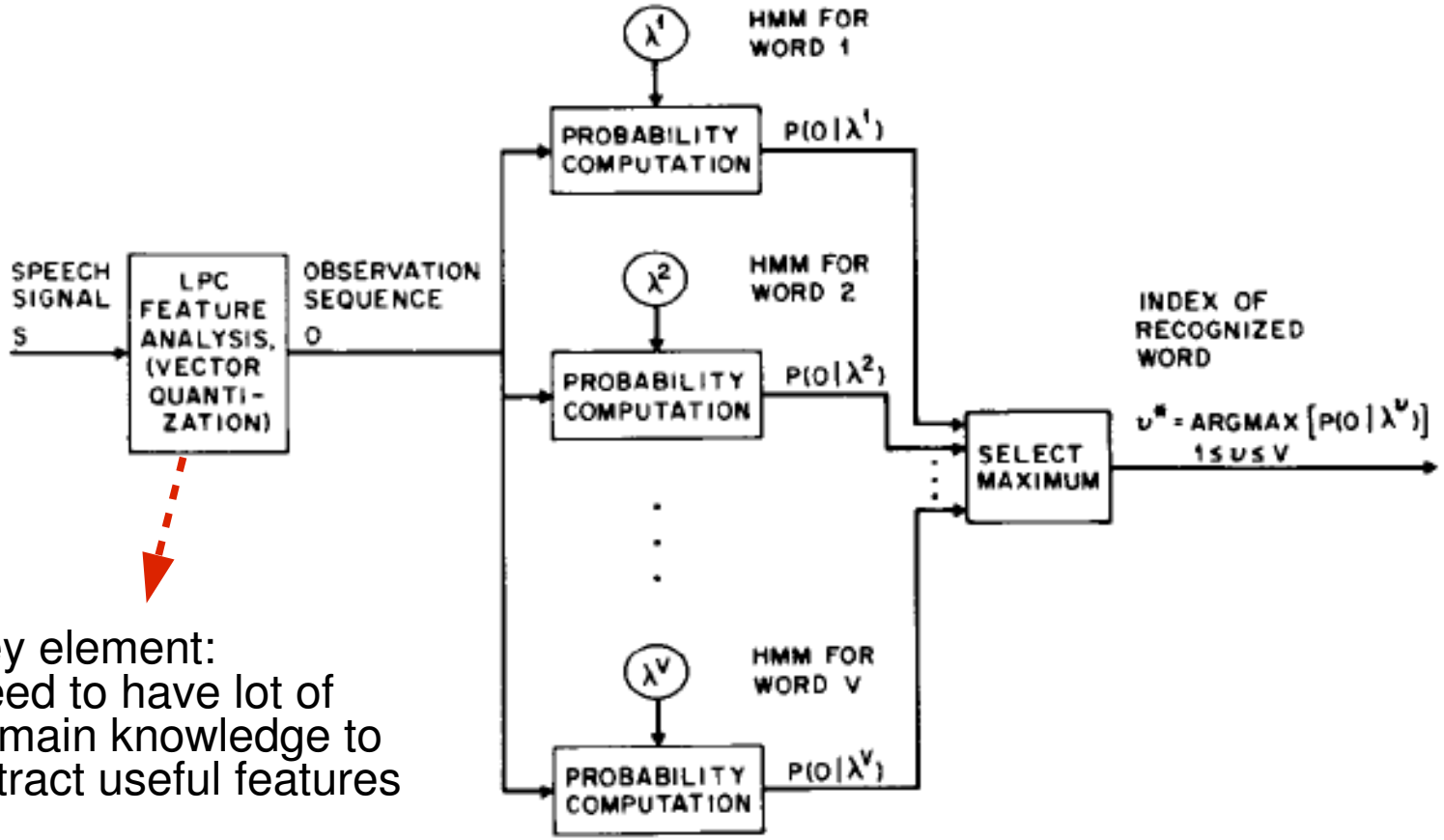
Composition of unit to have a word

Temporal and spectral analysis to obtain a observation sequence

Recognition of unit of the language. Word are divided into small unit to have a small set of models

Single word recognition

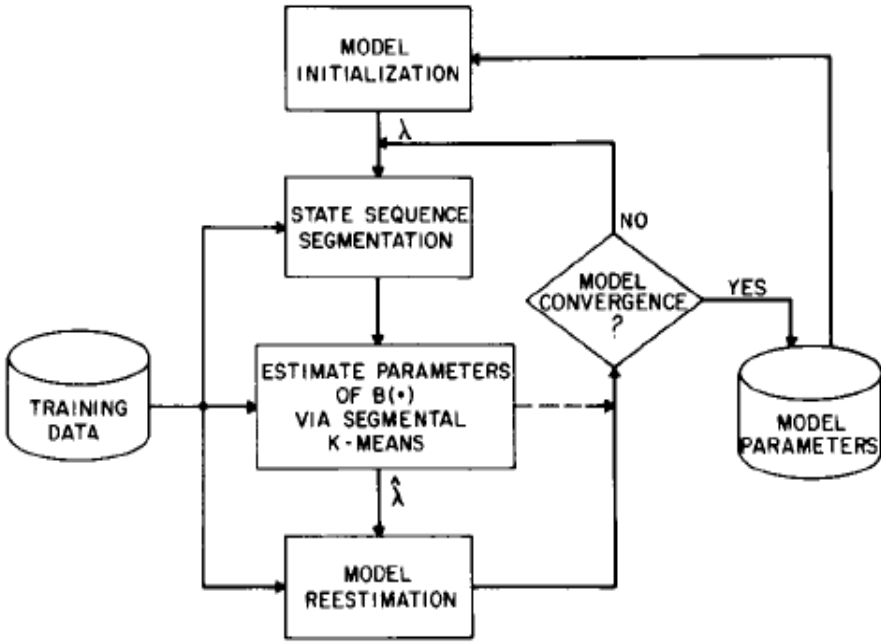
- Pronunciation of a single word an recognition of it

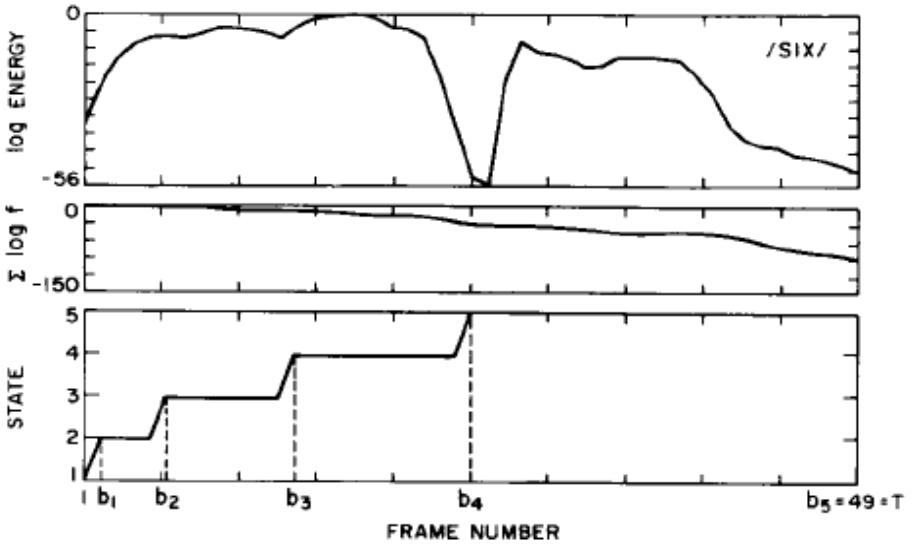


Key element:
Need to have lot of domain knowledge to extract useful features

Segmental K-Mean Segmentation into state

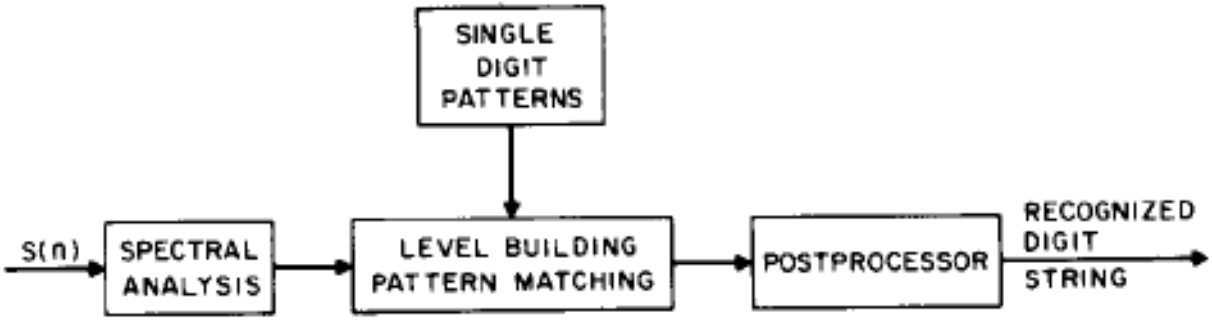
- When continuous observations is needed (i.e. speech recognition), the initial estimation of the observation distribution is important for the convergence





Recognition of a sequence of digits

- The observation sequence matched at each step with a single word recognition system



  - The observation sequences are non pre-classified and we have no information about the ending of each word
  - The building level match the observation sequence to a digit sequence with some probability
  - An alternative way is to use the state segmentation whit an higher level model

Conclusion

- HMMs are general stochastic models
- EM is a good algorithm to learn such models
- We need prior knowledge to define the structure of the model
- Lot of parameters needs lot of data
- They perform very well for many applications if they are applied in the correct way
  - Signal segmentation and classification
  - Clustering of signals
  - Prediction

Questions ?