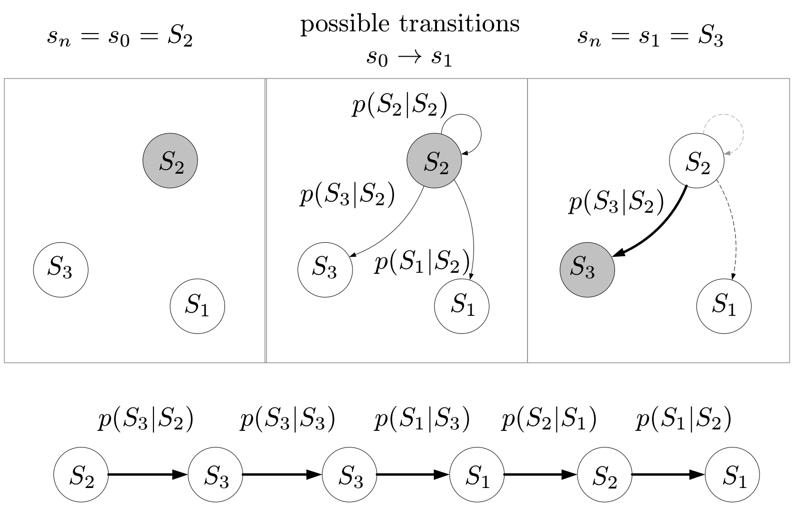
Hidden Markov Models

Kalle Ruttik

Discrete state based model

- **A Markov system has N states: $S_1, S_2, \ldots, S_N$**
- **At each discrete time moment the system is in one of the states: s_n**
- **Between the time steps the system could change the state**
  - **Transition to the next state is defined by the *transition probability***
- **Temporal behavior of the system can be described by the state sequence**
- **Probability of occurrence of a particular state sequence is the product of the initial state probability and the transition probabilities**

Example



Markov chain properties

Markov process (s_n can take continuous values)

- **A random process is a *Markov process* if the distribution of being in state s_n, given the infinite past, depends only on the previous state s_{n-1}**

Markov chain (finite state version of the Markov process)

- **A random process taking on only discrete possible values is a *Markov chain* if transition to the next state depends only on the current state**

$$p\left(s_n = S_j \mid s_{n-1} = S_{i_1}, s_{n-2} = S_{i_2}, \ldots\right) = p\left(s_n = S_j \mid s_{n-1} = S_{i_1}\right)$$

$p\left(s_n = S_j \mid s_{n-1} = S_{i_i}\right)$ **is state *transition probability***

Markov chain properties…

- If the next state depends on k previous states the model is called kth order HMM.

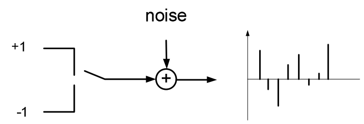
$$p\left(s_n = S_j \mid s_{n-1} = S_{i_1}, q_{n-2} = S_{i_2}, \ldots, s_{n-k} = S_{i_k}\right), \ \ 1 \le i_1, \ldots, i_k \le N$$

- Probability of any sequence of samples from a first order Markov chain can be evaluated as

$$\begin{aligned} & p\left(s_0, s_1, \ldots, s_{n-1}, s_n\right) \\ &= p\left(s_0\right) p\left(s_1 \mid s_0\right) p\left(s_2 \mid s_0, s_1\right) \ldots p\left(s_{n-1} \mid s_0, s_1, \ldots, s_{n-2}\right) p\left(s_n \mid s_0, s_1, \ldots, s_{n-1}\right) \\ &= p\left(s_0\right) p\left(s_1 \mid s_0\right) p\left(s_2 \mid s_1\right) \ldots p\left(s_{n-1} \mid s_{n-2}\right) p\left(s_n \mid s_{n-1}\right) \end{aligned}$$

Hidden Markov Model

- In Hidden Markov Model we can not observe the state sequence directly but only a process that depends on the state – the state is hidden
  - We observe a sequence of samples that is generated based on some particular state sequence



Example

Communication in additive noise. The symbols in message can take binary values +-1 (these are the states).
Received noisy message is the observed sequence.

How HMM can be used

- Given probability of an observed sequence find which sequence most likely produced it
  - Sequence is not known its probability is known
  - Used for language analysis
- Given the observed sequence(s) and the set of states what are the transition probabilities
  - Parameter estimation
- Given the observed sequence what is the most probable sequence of hidden states
  - Convolutional decoding and equalization

HMM state sequence estimation

- Maximum likelihood sequence estimation (MLSE)
  - Viterbi algorithm
  - Produces most probable state sequence
- Maximum a-posteriori probability (MAP)
  - BCJR algorithm for decoding
  - More complex than Viterbi decoder
  - Produces not only bit values but also reliability values for each bit

MLSE

- For estimating which sequence of states generated given observed sequence we have to
  - From the noisy observation calculate the probability of the state sequence
  - Calculate this probability for each possible state sequence
  - Select the sequence with highest probability – ML sequence
- The ML - estimate of the transmitted symbol sequence maximizes the joint probability density function of the observed signal conditioned by the state sequence

$$p\left(s_1, s_2, \ldots s_N \middle| y_1, y_2, \ldots y_N\right)$$

- Where $y = y_1, y_2, \ldots, y_n$ is the observed sequence

MLSE …

- The conditional probability that y was generated from a state sequence $s_1, s_2, \ldots, s_n$ can be calculated from Bayes' rule

$$p\left(s_1, s_2, \ldots s_N \middle| y_1, y_2, \ldots y_N\right) = \frac{p\left(y_1, y_2, \ldots y_N \middle| s_1, s_2, \ldots s_N\right) p\left(s_1, s_2, \ldots s_N\right)}{p\left(y_1, y_2, \ldots y_N\right)}$$

- If the added noise samples are independent the conditional signal samples are independent

$$p\left(y_1, y_2, \ldots y_N \middle| s_1, s_2, \ldots s_N\right) = \prod_{k=1}^{N} p\left(y_k \middle| s_1, s_2, \ldots s_N\right)$$

- Since yk is generated based on the state in time moment k (or in the transition from state k-1 to k) we can apply the Markov chain property $\prod_k p\left(y_k \middle| s_k s_{k-1}\right) p\left(s_k \middle| s_{k-1}\right)$
- the observed value at moment k depends only on the current state and it is independent on the previous states

MLSE…

- The ML probability should be evaluated for all the state sequences
  - Amount of them grows exponentially with the length of the sequence
- In Markov chain multiple sequences share a common part and the computation of ML can be simplified
- From the state model we can construct a trellis where each trellis stage corresponds to a time instant
- Each path trough the trellis corresponds to a state sequence
- Each state contributes to the probability a term $p\left(y_k \middle| s_k, s_{k-1}\right)$

MLSE…

- We can start to calculate the probability from one end of the trellis
- If two paths merge the one with higher probability corresponds to the state sequence with total higher probability
  - Because of the Markov property what follows is the same for both sequences
  - At each merge of the paths we need to remember only the path with higher probability
- At the end of the trellis we have N states with N remaining paths
- From those the one with highest probability is the ML sequence
- MLSE is an algorithm finding the path with a maximum probability (weight) trough the state trellis