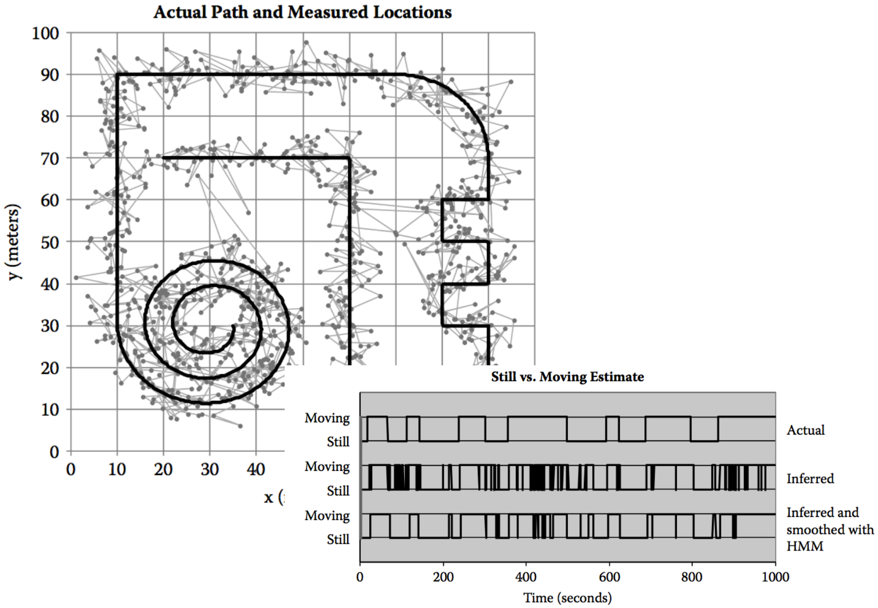
Markov Models for Pattern Recognition

— an introduction—

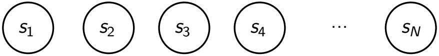
Thomas Plötz

November 2011



[taken from J Krumm (Ed.) "Ubiquitous Computing Fundamentatls"]

Hidden Markov Models: Two-Stage Stochastic Processes



1. Stage: discrete stochastic process $\hat{=}$ series of random variables which take on values from a discrete set of states (≈ finite automaton)

stationary: Process independent of absolute time t

causal: Distribution s_t only dependent on previous states

simple: *particularly* dependent only from *immediate* predecessor state ($\hat{=}$ first order)

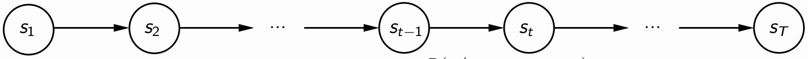
$\Rightarrow P(s_t|s_1, s_2, \ldots s_{t-1}) = P(s_t|s_{t-1})$

2. Stage: Depending on current state s_t for every point in time additionally an emission O_t is generated

$\Rightarrow P(O_t|O_1 \ldots O_{t-1}, s_1 \ldots s_t) = P(O_t|s_t)$

Caution: Only emissions can be observed → **hidden**

Hidden Markov Models: Two-Stage Stochastic Processes



1. Stage: discrete stochastic process $\hat{=}$ series of random variables which take on values from a discrete set of states ($\approx$ finite automaton)

   stationary: Process independent of absolute time t

   causal: Distribution s_t only dependent on previous states

   simple: *particularly* dependent only from *immediate* predecessor state ($\hat{=}$ first order)

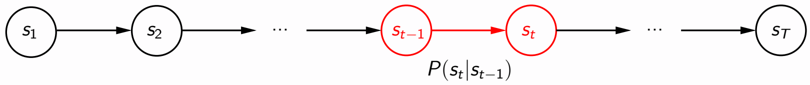
   $\Rightarrow P(s_t|s_1, s_2, \ldots s_{t-1}) = P(s_t|s_{t-1})$

2. Stage: Depending on current state s_t for every point in time additionally an emission O_t is generated

   $\Rightarrow P(O_t|O_1 \ldots O_{t-1}, s_1 \ldots s_t) = P(O_t|s_t)$

   Caution: Only emissions can be observed $\rightarrow$ **hidden**

Hidden Markov Models: Two-Stage Stochastic Processes



1. Stage: discrete stochastic process $\hat{=}$ series of random variables which take on values from a discrete set of states (≈ finite automaton)

   stationary: Process independent of absolute time t

   causal: Distribution s_t only dependent on previous states

   simple: *particularly* dependent only from *immediate* predecessor state ($\hat{=}$ first order)

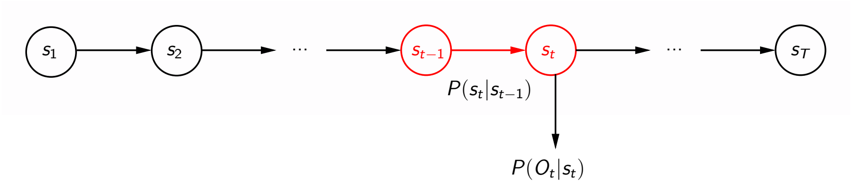
   $\Rightarrow P(s_t|s_1, s_2, \ldots s_{t-1}) = P(s_t|s_{t-1})$

2. Stage: Depending on current state s_t for every point in time additionally an emission O_t is generated

   $\Rightarrow P(O_t|O_1 \ldots O_{t-1}, s_1 \ldots s_t) = P(O_t|s_t)$

   Caution: Only emissions can be observed → **hidden**

Hidden Markov Models: Two-Stage Stochastic Processes



1. Stage: discrete stochastic process $\hat{=}$ series of random variables which take on values from a discrete set of states ($\approx$ finite automaton)

   stationary: Process independent of absolute time t

   causal: Distribution s_t only dependent on previous states

   simple: *particularly* dependent only from *immediate* predecessor state ($\hat{=}$ first order)

   $\Rightarrow P(s_t|s_1, s_2, \ldots s_{t-1}) = P(s_t|s_{t-1})$

2. Stage: Depending on current state s_t for every point in time additionally an emission O_t is generated

   $\Rightarrow P(O_t|O_1 \ldots O_{t-1}, s_1 \ldots s_t) = P(O_t|s_t)$

   Caution: Only emissions can be observed $\rightarrow$ **hidden**

Hidden-Markov-Models: Formal Definition

A Hidden-Markov-Model λ of *first order* is defined as:

- a finite set of states:

$$\{s | 1 \leq s \leq N\}$$

- a matrix of state transition probabilities:

$$\mathbf{A} = \{a_{ij} | a_{ij} = P(s_t = j | s_{t-1} = i)\}$$

- a vector of start probabilities:

$$\boldsymbol{\pi} = \{\pi_i | \pi_i = P(s_1 = i)\}$$

- state specific emission probability distributions:

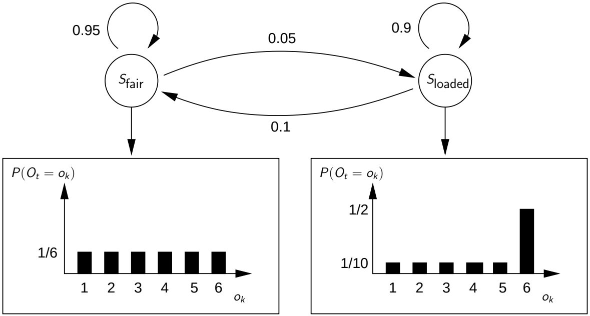
$$\mathbf{B} = \{b_{jk} | b_{jk} = P(O_t = o_k | s_t = j)\} \text{ (discrete case)}$$

or

$$\{b_j(O_t) | b_j(O_t) = p(O_t | s_t = j)\} \text{ (continuous case)}$$

Toy Example: The Occasionally Dishonest Casino – I

[idea from [?]]



$O = 1, 1, 2, 6, 6, 6, 3, 5$

Background: Casino occasionally exchanging dice: fair ⇔ loaded
⇒ Model with two states: S_{fair} and S_{loaded}

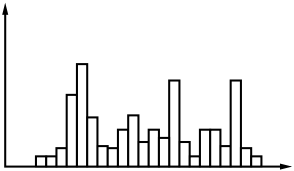
Exclusive observations: Results of the rolls
⇒ Underlying state-sequence remains hidden!

Question: Which die has been used, i.e. when is the casino cheating?
⇒ Probabilistic inference about internal state-sequence using stochastic model

Modeling of Emissions

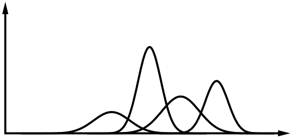
Discrete inventory of symbols: Very limited application fields

- ✓ Well suited for discrete data (e.g. DNA)
- ϟ Inappropriate for non-discrete data – use of vector quantizer required!



Continuous modeling: Standard for most pattern recognition applications processing sensory data

- ✓ Treatment of real-valued vector data (i.e. vast majority of "real-world" data)
- ✓ Defines distributions over $\mathbb{R}^n$



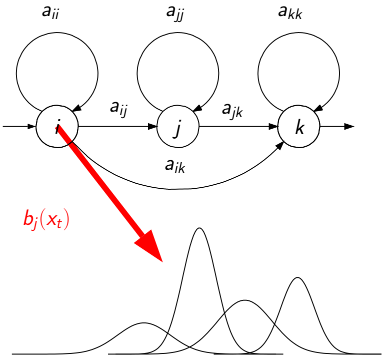
Problem: No general parametric description

Procedure: Approximation using mixture densities

$$p(\mathbf{x}) \;\hat{=}\; \sum_{k=1}^{\infty} c_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{C}_k)$$

$$\approx \sum_{k=1}^{M} c_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \mathbf{C}_k)$$

Modeling of Emissions – II

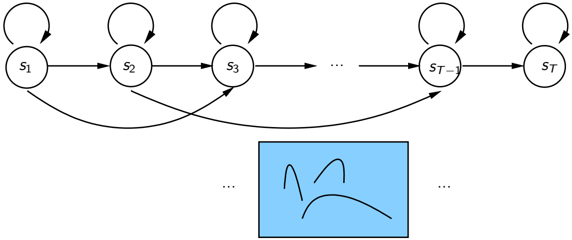


Mixture density modeling:

- Base Distribution?
  ⇒ Gaussian Normal densities
- Shape of Distributions (full / diagonal covariances)?
  ⇒ Depends on pre-processing of the data (e.g. redundancy reduction)
- Number of mixtures?
  ⇒ Clustering (. . . and heuristics)
- Estimation of mixtures?
  ⇒ e.g. Expectation-Maximization

[↗ Practice]

Usage Concepts for Hidden-Markov-Models



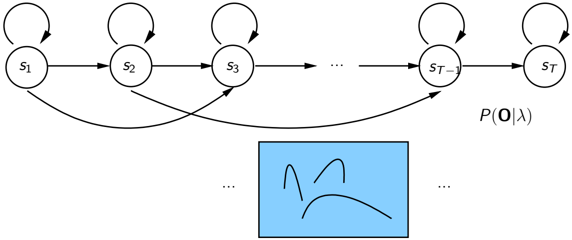
Assumption: Patterns observed are generated by stochastic models which are comparable *in principle*

Scoring: How well describes the model some pattern?
$\longrightarrow$ Determination of the production probability $P(\mathbf{O}|\lambda)$

Decoding: What is the "internal structure" of the model? ($\hat{=}$ "Recognition")
$\longrightarrow$ Determination of the optimal state sequence
$$\mathbf{s}^* = \underset{\mathbf{s}}{\operatorname{argmax}} \, P(\mathbf{O}, \mathbf{s}|\lambda)$$

Training: How to determine the "optimal" model?
$\longrightarrow$ Improvement of a given model λ with $P(\mathbf{O}|\hat{\lambda}) \geq P(\mathbf{O}|\lambda)$

Usage Concepts for Hidden-Markov-Models



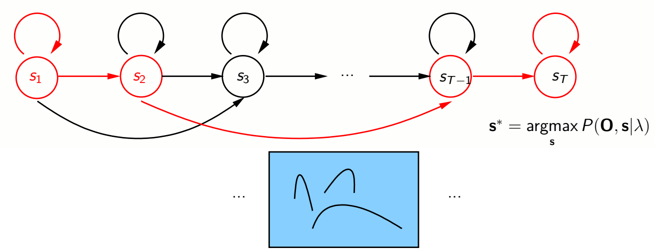
Assumption: Patterns observed are generated by stochastic models which are comparable *in principle*

Scoring: How well describes the model some pattern?
$\rightarrow$ Determination of the production probability $P(\mathbf{O}|\lambda)$

Decoding: What is the "internal structure" of the model? ($\hat{=}$ "Recognition")
$\rightarrow$ Determination of the optimal state sequence
$\mathbf{s}^* = \underset{\mathbf{s}}{\operatorname{argmax}} P(\mathbf{O}, \mathbf{s}|\lambda)$

Training: How to determine the "optimal" model?
$\rightarrow$ Improvement of a given model λ with $P(\mathbf{O}|\hat{\lambda}) \geq P(\mathbf{O}|\lambda)$

Usage Concepts for Hidden-Markov-Models



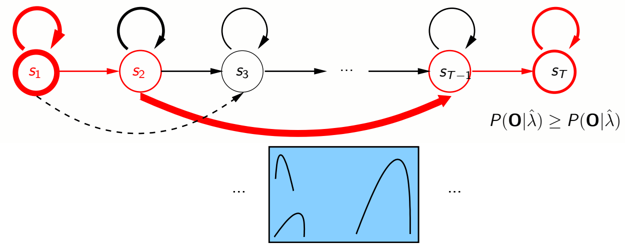
Assumption: Patterns observed are generated by stochastic models which are comparable *in principle*

Scoring: How well describes the model some pattern?
$\rightarrow$ Determination of the production probability $P(\mathbf{O}|\lambda)$

Decoding: What is the "internal structure" of the model? ($\hat{=}$ "Recognition")
$\rightarrow$ Determination of the optimal state sequence
$\mathbf{s}^* = \underset{\mathbf{s}}{\text{argmax}}\, P(\mathbf{O}, \mathbf{s}|\lambda)$

Training: How to determine the "optimal" model?
$\rightarrow$ Improvement of a given model λ with $P(\mathbf{O}|\hat{\lambda}) \geq P(\mathbf{O}|\lambda)$

Usage Concepts for Hidden-Markov-Models



Assumption: Patterns observed are generated by stochastic models which are comparable *in principle*

Scoring: How well describes the model some pattern?
$\rightarrow$ Determination of the production probability $P(\mathbf{O}|\lambda)$

Decoding: What is the "internal structure" of the model? ($\hat{=}$ "Recognition")
$\rightarrow$ Determination of the optimal state sequence

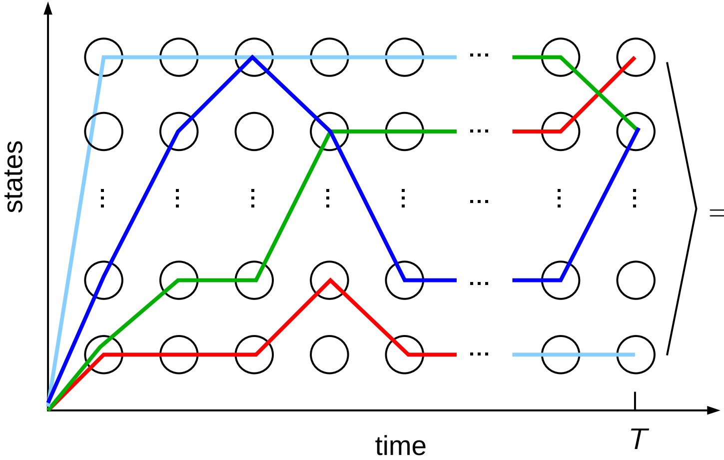
$$\mathbf{s}^* = \underset{\mathbf{s}}{\operatorname{argmax}}\, P(\mathbf{O}, \mathbf{s}|\lambda)$$

Training: How to determine the "optimal" model?
$\rightarrow$ Improvement of a given model λ with $P(\mathbf{O}|\hat{\lambda}) \geq P(\mathbf{O}|\lambda)$

The Production Probability

Wanted: Assessment of HMMs' quality for describing statistical properties of data

Widely used measure: *Production probability* $P(\mathbf{O}|\lambda)$ that observation sequence $\mathbf{O}$ was generated by model λ – along an arbitrary state sequence



$\Rightarrow P(\mathbf{O}|\lambda)$

The Production Probability: Naive Determination

1. Probability for generating observation sequence $O_1, O_2, \ldots O_T$ along corresponding state sequence $\mathbf{s} = s_1, s_2, \ldots s_T$ of same length:

$$P(\mathbf{O}|\mathbf{s}, \lambda) = \prod_{t=1}^{T} b_{s_t}(O_t)$$

2. Probability that a given model λ runs through arbitrary state sequence:

$$P(\mathbf{s}|\lambda) = \pi_{s_1} \prod_{t=2}^{T} a_{s_{t-1},s_t} = \prod_{t=1}^{T} a_{s_{t-1},s_t}$$

3. (1) + (2): Probability that λ generates $\mathbf{O}$ along certain state sequence $\mathbf{s}$:

$$P(\mathbf{O}, \mathbf{s}|\lambda) = P(\mathbf{O}|\mathbf{s}, \lambda)P(\mathbf{s}|\lambda) = \prod_{t=1}^{T} a_{s_{t-1},s_t} b_{s_t}(O_t)$$

4. Total $P(\mathbf{O}|\lambda)$: Summation over all possible state sequences of length T

$$P(\mathbf{O}|\lambda) = \sum_{\mathbf{s}} P(\mathbf{O}, \mathbf{s}|\lambda) = \sum_{\mathbf{s}} P(\mathbf{O}|\mathbf{s}, \lambda)P(\mathbf{s}|\lambda)$$

ϟ Complexity: $O(TN^T)$

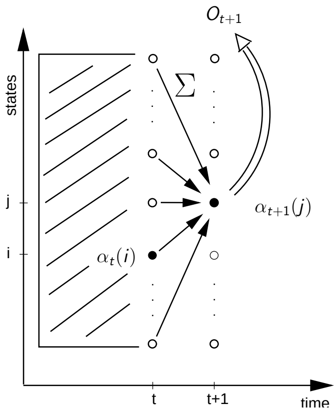
The Production Probability: The Forward-Algorithm

More efficient: Exploitation of the Markov-property, i.e. the "finite memory"
⇒ "Decisions" only dependent on immediate predecessor state

Let:
$\alpha_t(i) = P(O_1, O_2, \ldots O_t, s_t = i|\lambda)$
(*forward variable*)

1. $\alpha_1(i) := \pi_i b_i(O_1)$
2. $\alpha_{t+1}(j) := \left\{ \sum_{i=1}^{N} \alpha_t(i) a_{ij} \right\} b_j(O_{t+1})$
3. $P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$

✓ Complexity: $O(TN^2)$!
(vs. $O(TN^T)$ from naive determination)



Later: Backward-Algorithm [↗ Training]

The "optimal" Production Probability

Total production probability: Consider *all* paths through model

- ✓ Mathematically exact determination of $P(\mathbf{O}|\lambda)$
- Specialization of partial models within total model cannot be judged

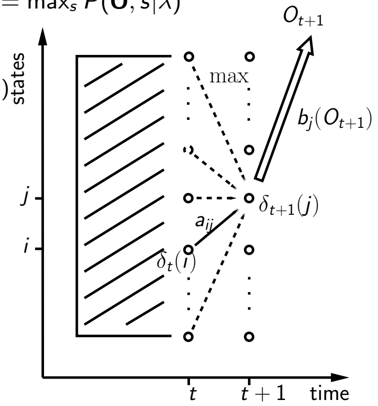
Modification: Consider only respective optimal possibility to generate $\mathbf{O}$, given λ

- ✓ Discrimination between λ_1 (satisfying on average) / λ_2 (specialized)

Optimal probability: $P^*(\mathbf{O}|\lambda) = P(\mathbf{O}, s^*|\lambda) = \max_s P(\mathbf{O}, s|\lambda)$

$$\delta_t(i) = \max_{s_1,\dots s_{t-1}} P(O_1, \dots O_t, s_1, \dots s_{t-1}, s_t = i|\lambda)$$

1. $\delta_1(i) = \pi_i b_i(O_1)$
2. $\forall t,\ t = 1 \dots T-1$:
   $\delta_{t+1}(j) = \max_i \{\delta_t(i) a_{ij}\}\, b_j(O_{t+1})$
3. $P^*(\mathbf{O}|\lambda) = P(\mathbf{O}, \mathbf{s}^*|\lambda) = \max_i \delta_T(i)$



Decoding

Problem: Global production probability $P(\mathbf{O}|\lambda)$ not sufficient for analysis if individual states are associated to meaningful segments of data

$\Rightarrow$ (Probabilistic) Determination of optimal state sequence $\mathbf{s}^*$ necessary

Maximization of posterior probability:

$$\mathbf{s}^* = \underset{\mathbf{s}}{\operatorname{argmax}}\, P(\mathbf{s}|\mathbf{O}, \lambda)$$

Bayes' rule:

$$P(\mathbf{s}|\mathbf{O}, \lambda) = \frac{P(\mathbf{O}, \mathbf{s}|\lambda)}{P(\mathbf{O}|\lambda)}$$

$P(\mathbf{O}|\lambda)$ irrelevant (constant for fixed $\mathbf{O}$ and given λ), thus:

$$\mathbf{s}^* = \underset{\mathbf{s}}{\operatorname{argmax}}\, P(\mathbf{s}|\mathbf{O}, \lambda) = \underset{\mathbf{s}}{\operatorname{argmax}}\, P(\mathbf{O}, \mathbf{s}|\lambda)$$

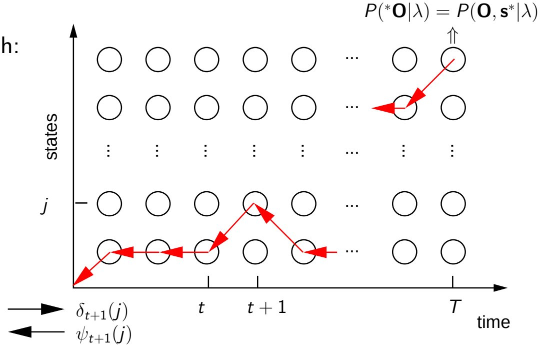
Determination of $\mathbf{s}^*$: Brute-Force [↗ Optimal Production Probability] or more efficiently: *Viterbi-Algorithm*

The Viterbi Algorithm

...inductive procedure for efficient determination of $\mathbf{s}^*$ exploiting Markov property

Let: $\delta_t(i) = \max\limits_{s_1,s_2,\ldots s_{t-1}} P(O_1, O_2, \ldots O_t, s_t = i|\lambda)$

1. $\delta_1(i) := \pi_i b_i(O_1)$ $\qquad \psi_1(i) := 0$
2. $\delta_{t+1}(j) := \max\limits_i(\delta_t(i)a_{ij})b_j(O_{t+1})$ $\qquad \psi_{t+1}(j) := \operatorname*{argmax}\limits_i \ldots$
3. $P^*(\mathbf{O}|\lambda) = P(\mathbf{O}, \mathbf{s}^*|\lambda) = \max\limits_i \delta_T(i)$
   $s_T^* := \operatorname*{argmax}\limits_j \delta_T(j)$
4. Back-tracking of optimal path:
   $s_t^* = \psi_{t+1}(s_{t+1}^*)$

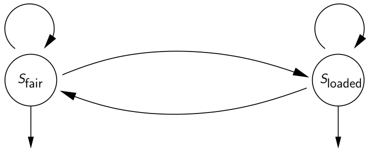


✓ Implicit *segmentation*

✓ Linear complexity in time

STOP Quadratic complexity w.r.t. #states

Toy Example: The Occasionally Dishonest Casino – II



Parameters of the given HMM λ:

- Start probabilities: $\boldsymbol{\pi} = (1/2 \quad 1/2)^T$
- Transition probabilities: $\mathbf{A} = \begin{pmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{pmatrix}$
- Emission probabilities: $\mathbf{B} = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \end{pmatrix}$
- Observation sequence: $\mathbf{O} = O_1, O_2, \ldots, O_T = 1, 1, 2, 6, 6, 6, 3, 5$

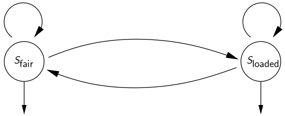
Wanted: Internal state-sequence for segmentation into fair use and cheating
$\Rightarrow$ Viterbi-Algorithm

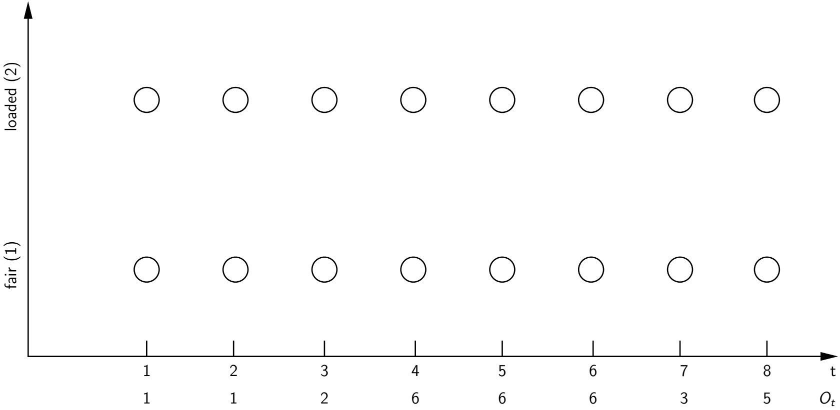
Toy Example: The Occasionally Dishonest Casino – III

$$\pi_i = 1/2, \quad \mathbf{A} = \begin{pmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \end{pmatrix}$$

$$\mathbf{O} = 1, 1, 2, 6, 6, 6, 3, 5$$



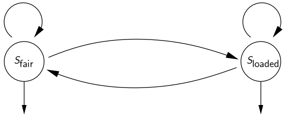


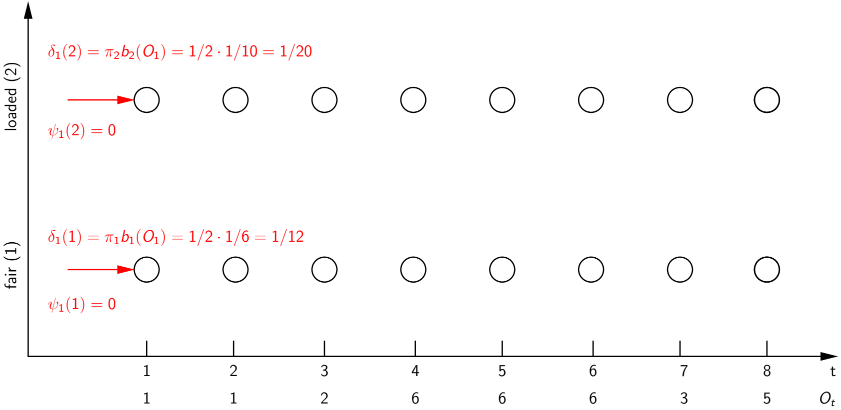
Toy Example: The Occasionally Dishonest Casino – III

$$\pi_i = 1/2, \quad \mathbf{A} = \begin{pmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \end{pmatrix}$$

$$\mathbf{O} = 1, 1, 2, 6, 6, 6, 3, 5$$



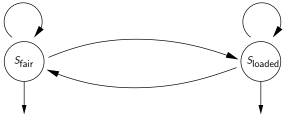


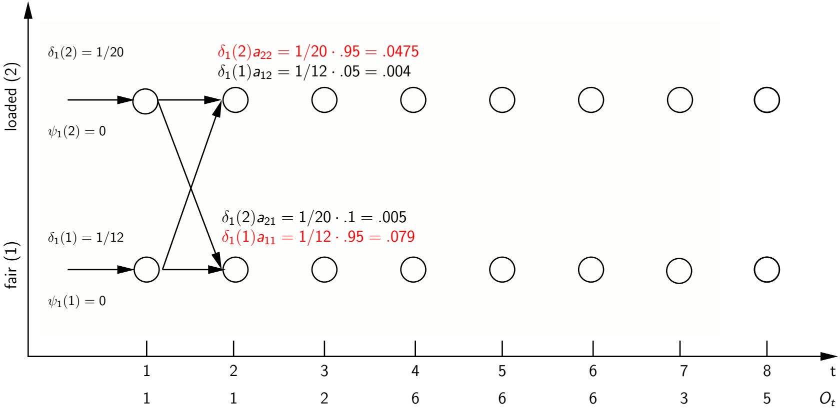
Toy Example: The Occasionally Dishonest Casino – III

$$\pi_i = 1/2, \quad \mathbf{A} = \begin{pmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \end{pmatrix}$$

$$\mathbf{O} = 1, 1, 2, 6, 6, 6, 3, 5$$



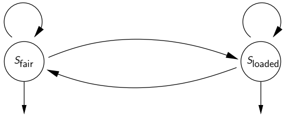


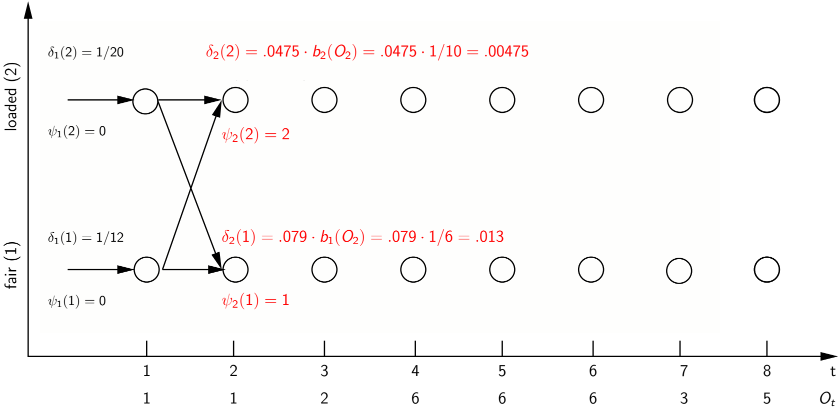
Toy Example: The Occasionally Dishonest Casino – III

$$\pi_i = 1/2, \quad \mathbf{A} = \begin{pmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \end{pmatrix}$$

$$\mathbf{O} = 1, 1, 2, 6, 6, 6, 3, 5$$



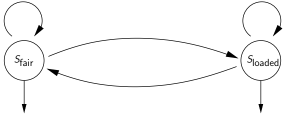


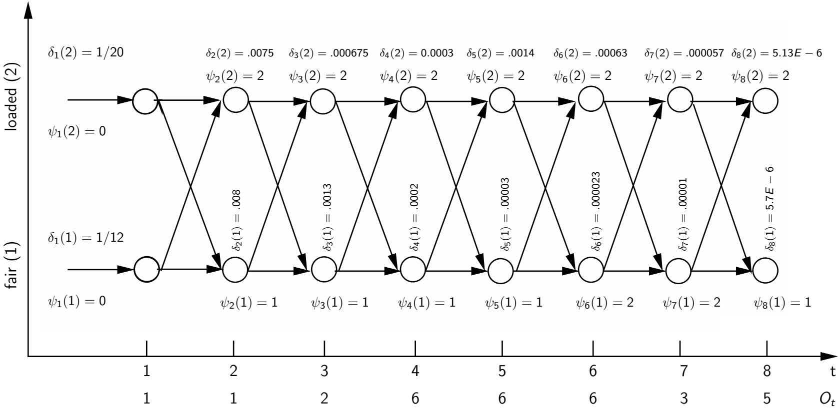
Toy Example: The Occasionally Dishonest Casino – III

$$\pi_i = 1/2, \quad \mathbf{A} = \begin{pmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \end{pmatrix}$$

$$\mathbf{O} = 1, 1, 2, 6, 6, 6, 3, 5$$



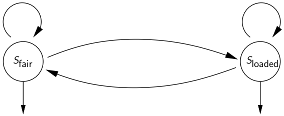


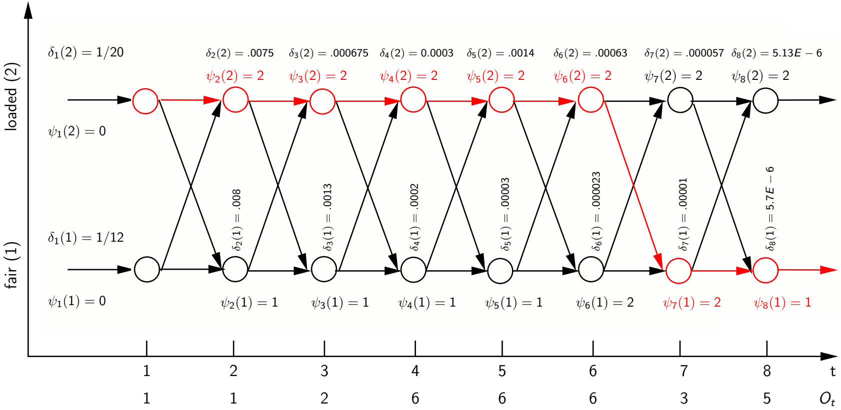
Toy Example: The Occasionally Dishonest Casino – III

$$\pi_i = 1/2, \quad \mathbf{A} = \begin{pmatrix} 0.95 & 0.05 \\ 0.1 & 0.9 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \end{pmatrix}$$

$$\mathbf{O} = 1, 1, 2, 6, 6, 6, 3, 5$$





Parameter Estimation – Fundamentals

Goal: Derive optimal (for some purpose) statistical model from sample data

Problem: No suitable analytical method / algorithm known

"Work-Around": Iteratively improve existing model λ
$\Rightarrow$ Optimized model $\hat{\lambda}$ better suited for given sample data

General procedure: Parameters of λ subject to growth transformation such that

$$P(\ldots|\hat{\lambda}) \geq P(\ldots|\lambda)$$

1. "Observe" model's actions during generation of an observation sequence
2. Original parameters are replaced by relative frequencies of respective events

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from } i \text{ to } j}{\text{expected number of transitions out of state } i}$$

$$\hat{b}_i(o_k) = \frac{\text{expected number of outputs of } o_k \text{ in state } i}{\text{total number of outputs in state } i}$$

- Only probabilistic inference of events possible!
- (Posterior) state probability required!

The Posterior State Probability

Goal: Efficiently compute $P(S_t = i|\mathbf{O}, \lambda)$ for model assessment

Procedure: Exploit limited memory for

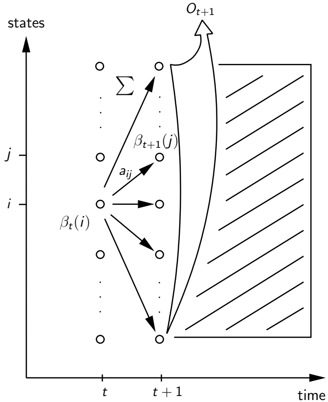
- History – forward-probability $\alpha_t(i)$ [↗ forward-algorithm], and
- Rest of partial observation sequence – *backward-probability* $\beta_t(i)$

Backward-Algorithm:

Let

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \ldots O_T|s_t = i, \lambda)$$

1. $\beta_T(i) := 1$
2. For all times t, $t = T-1 \ldots 1$:
   $\beta_t(i) := \sum_j a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)$
3. $P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \pi_i b_i(O_1) \beta_1(i)$



The Forward-Backward Algorithm

... for efficient determination of posterior state probability

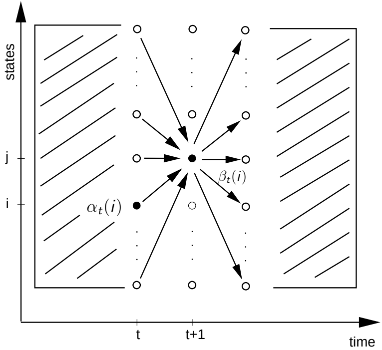
$$P(S_t = i|\mathbf{O}, \lambda) = \frac{P(S_t = i, \mathbf{O}|\lambda)}{P(\mathbf{O}|\lambda)}$$

[↗ forward-algorithm]

$$P(S_t = i, \mathbf{O}|\lambda) = P(O_1, O_2, \ldots O_t, S_t = i|\lambda) P(O_{t+1}, O_{t+2}, \ldots O_T|S_t = i, \lambda)$$

$$= \alpha_t(i)\beta_t(i)$$

$$\Rightarrow \quad \gamma_t(i) = P(S_t = i|\mathbf{O}, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(\mathbf{O}|\lambda)}$$



Parameter Training using the Baum-Welch-Algorithm

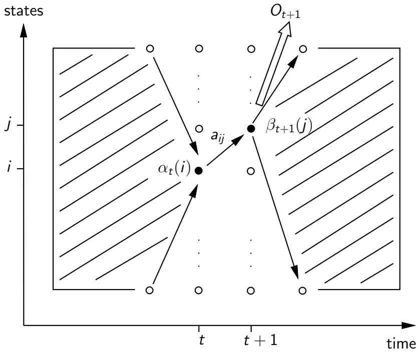
Background: Variant of *Expectation Maximization (EM)*-Algorithm
(parameter estimation for stochastic models incl. hidden random variables)

Optimization criterion: Total production probability $P(\mathbf{O}|\lambda)$, thus

$$P(\mathbf{O}|\hat{\lambda}) \geq P(\mathbf{O}|\lambda)$$

Definitions: (of quantities based on forward- and backward variables)
$\Rightarrow$ Allow (statistical) inferences about internal processes of λ when generating $\mathbf{O}$

$$\begin{aligned}
\gamma_t(i,j) &= P(S_t = i, S_{t+1} = j|\mathbf{O}, \lambda) \\
&= \frac{P(S_t = i, S_{t+1} = j, \mathbf{O}|\lambda)}{P(\mathbf{O}|\lambda)} \\
&= \frac{\alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}{P(\mathbf{O}|\lambda)} \\
\gamma_t(i) &= P(S_t = i|\mathbf{O}, \lambda) \\
&= \sum_{j=1}^{N} P(S_t = i, S_{t+1} = j|\mathbf{O}, \lambda) \\
&= \sum_{j=1}^{N} \gamma_t(i,j)
\end{aligned}$$



The Baum-Welch-Algorithm

Let

$$\gamma_t(i) = P(S_t = i|\mathbf{O}, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(\mathbf{O}|\lambda)}$$

$$\gamma_t(i,j) = P(S_t = i, S_{t+1} = j|\mathbf{O}, \lambda) = \frac{\alpha_t(i)\, a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j)}{P(\mathbf{O}|\lambda)}$$

$$\xi_t(j,k) = P(S_t = j, M_t = k|\mathbf{O}, \lambda) = \frac{\sum_{i=1}^{N} \alpha_t(i)\, a_{ij}\, c_{jk}\, g_{jk}(O_t)\, \beta_t(j)}{P(\mathbf{O}|\lambda)}$$

1. Choose a suitable initial model $\lambda = (\boldsymbol{\pi}, \mathbf{A}, \mathbf{B})$ with initial estimates (π_i, a_{ij}, c_{jk} for mixtures $g_{jk}(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{jk}, \mathbf{C}_{jk})$ for pdf. $b_{jk}(\mathbf{x}) = \sum_k c_{jk}\, g_{jk}(\mathbf{x})$.)

2. Compute updated estimates $\hat{\lambda} = (\hat{\boldsymbol{\pi}}, \hat{\mathbf{A}}, \hat{\mathbf{B}})$ for all model parameters:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \gamma_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \qquad \hat{\pi}_i = \gamma_1(i)$$

$$\hat{c}_{jk} = \frac{\sum_{t=1}^{T} \xi_t(j,k)}{\sum_{t=1}^{T} \gamma_t(j)}$$

$$\hat{\boldsymbol{\mu}}_{jk} = \frac{\sum_{t=1}^{T} \xi_t(j,k)\, \mathbf{x}_t}{\sum_{t=1}^{T} \xi_t(j,k)} \qquad \hat{\mathbf{C}}_{jk} = \frac{\sum_{t=1}^{T} \xi_t(j,k)\, \mathbf{x}_t \mathbf{x}_t^T}{\sum_{t=1}^{T} \xi_t(j,k)} - \hat{\boldsymbol{\mu}}_{jk} \hat{\boldsymbol{\mu}}_{jk}^T$$

3. **if** $P(\mathbf{O}|\hat{\lambda})$ was considerably improved by the updated model $\hat{\lambda}$ w.r.t. λ

   let $\lambda \leftarrow \hat{\lambda}$ and continue with step 2

   otherwise Stop!

Multiple Observation Sequences

In general: Sample sets used for parameter training are subdivided into individual segments – so-called *turns*

So far: Turns were considered individual observation sequences

Goal: Estimate model parameters also on a *set* of isolated sequences

Procedure: Accumulate across all observation sequences considered statistics gathered for the updating of the parameters

Example:

$$\hat{\boldsymbol{\mu}}_{jk} = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T} \xi_t^l(j,k)\, \mathbf{x}_t}{\sum_{l=1}^{L} \sum_{t=1}^{T} \xi_t^l(j,k)}$$

Hidden Markov Models: Summary

Pros and Cons:

- ✓ Two-stage stochastic process for analysis of highly variant patterns
  (allows for probabilistic inference about internal state sequence – i.e. recognition)
- ✓ Efficient algorithms for training and evaluation, resp., exist
  (Forward-Backward, Viterbi-decoding, Baum-Welch)
- ✓ Can "easily" be combined with statistical language model
  (channel model: integration of [↗ Markov chain models])
- ↯ Considerable amounts of training data necessary
  ("There's no data like more data!" [?])

Variants and Extensions (not covered *here*):

- Hybrid models increased robustness
  (often combination with neural networks)
- Techniques for fast and robust adaptation, i.e. specialization, exist
  (Maximum A-posteriori adaptation, Maximum Likelihood Linear Regression)