Markovian models for pattern recognition

Václav Hlaváč

Czech Technical University in Prague
Czech Institute of Informatics, Robotics and Cybernetics
160 00 Prague 6, Jugoslávských partyzánů 1580/3, Czech Republic
http://people.ciirc.cvut.cz/hlavac, vaclav.hlavac@cvut.cz

Courtesy: M.I. Schlesinger

Outline of the talk:

- Motivation, use of.
- Stochastic finite automata.
- Markovian statistical model.
- Three most common tasks with hidden Markovian models (recognition, seeking the most probable sequence of hidden states, learning markovian statistical models empirically).

Context-based classification

Hidden Markov sequence (also chain) statistical model, abbreviated HMM

- There is a sequence of decisions instead of a single decision. Next decision depends on previous decisions.
- Usual applications: analysis of observations changing in time. E.g., the speech signal, time sequence of strokes in handwriting.
- Hidden Markov Model for sequences constitutes the "golden standard" in time series analysis.
- What is the reason?

  The Hidden Markov sequence model is the most complex statistical model, for which there is a polynomial complexity algorithm (dynamic programming algorithm).

Hidden Markov field constitutes the next simplest statistical model for grid-like structures (e.g., pixels in images). No polynomial complexity algorithms are available any more.

Notation of sequences simplifying expressions

- While describing HMMs, we will deal with sequences of observations $\bar{x}$ and with sequences of hidden states $\bar{y}$.

$$\begin{array}{lll} \text{Observations} & \bar{x} = (& x_1, x_2, \ldots, x_n) \quad \in X^n \\ \text{Hidden states} & \bar{y} = (y_0, & y_1, y_2, \ldots, y_n) \quad \in Y^{n+1} \end{array}$$

- Let introduce a more concise sequences notation: $\bar{x} = (x_a, x_{a+1}, \ldots, x_b) = x_a^b$, which will simplify expressions.
- For instance, the sequence of observations $\bar{x}$ and the sequence of hidden states $\bar{y}$ introduced above simplifies to

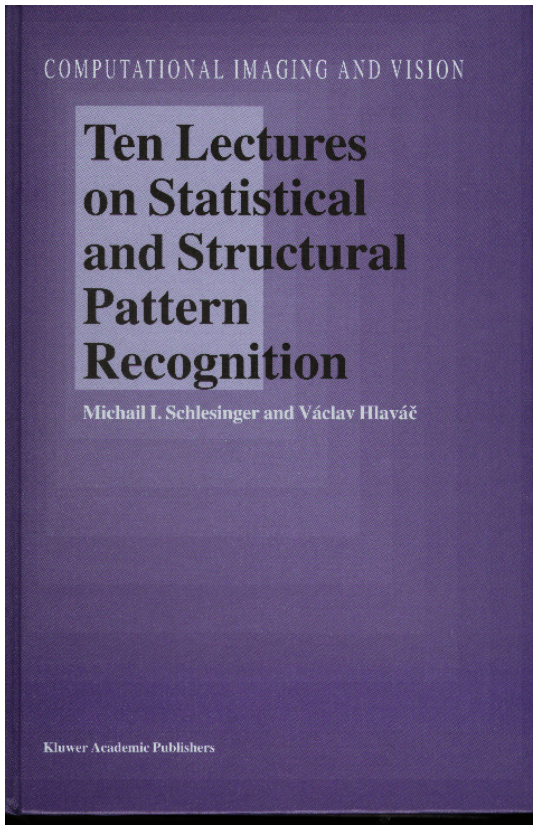
$$\begin{array}{ll} \text{Observations (conciselly)} & \bar{x} = x_1^n \\ \text{Hidden states (conciselly)} & \bar{y} = y_0^n \end{array}$$

Examples of application areas

- Speech signal recognition ($\bar{x}$ is the signal from a microphone, $\bar{y}$ phonemes).
- Seeking word(s) in the utterance ($\bar{x}$ sequence of words, $\bar{y}$ target word(s)).
- Recognition of handwritten characters/symbols, ($\bar{x}$ on-line strokes of the pen, $\bar{y}$, e.g., individual target characters, signatures).
- Biomedical engineering, e.g., ECG, EEG signal analysis, ($\bar{x}$ signal, $\bar{y}$ features of a signal).
- Bioinformatics, e.g. DNA sequences analysis ($\bar{x}$ responses of fluorescence-marked molecules, $y \in \{A, C, G, T\}$) or ($x \in \{A, C, G, T\}$, $\bar{y}$ subsequences interesting from the interpretation point of view).
- Mobile robotics ($\bar{x}$ points on a robot trajectory, $\bar{y}$ trajectory interpretation).
- Recognition in images. However, a special case can be treated only, which enables one-dimensional ordering. E.g., recognition of car number plates. ($\bar{x}$ columns of the car numberplate image, $\bar{y}$ characters and symbols used on a car numberplate).

Recommended reading

- Schlesinger M.I., Hlaváč V.: Ten Lectures on Statistical and Structural Pattern Recognition, Kluwer Academic Publishers, Dordrecht 2002
- Rabiner L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition, Journal Proceedings of the IEEE, Vol. 77, No. 2, 1989, pp. 257-286.



Andrey Andrejevich Markov

- Born in Ryazan, Russia 1856, died 1922 in St. Petersburg.
- Russian mathematician, researched also stochastic processes; Professor of the St. Petersburg University, member of the Russian Academy of Science.
- Markov property on chains: Sequences of random variables, the value of the next variable is determined by the value of the previous variable but independent on previous states.
- Andrey Markov used Markov chains (paper from 1912) to study the distribution of vowels in Eugene Onegin poem by Alexander Pushkin. He proved the central limit theorem for such chains.

Markov models and automata

- Markov models (including hidden ones, HMMs) are a special instance of stochastic finite automata.
- Markov models enable expressing statistical dependencies given by the order of observations (states) as, e.g., in time series.

Finite automaton $(Y, V, X, \delta, k_0, F)$

- Y - a finite set of automaton states (hidden states);
- V - a finite alphabet of input symbols;
- X - a finite alphabet of output symbols;
- y_0 - an initial (hidden) state, $y_0 \in Y$;
- F - target states; $F \subset Y$;
- δ - a state transition function; $\delta\colon Y \times V \to Y \times X$.

- If the automaton is in the state $y \in Y$ and the symbol $v \in V$ is brought to its input, the automaton changes to state $y' \in Y$, and generates the output symbol $x \in X$.
- The transition function δ determines the tuple $(y', x) = \delta(y, v)$.
- The automaton operates iteratively until a state $y' \in F$ is attained, which is the stopping condition.

Stochastic finite automaton

- We will generalize the finite automaton in such a way that transitions from a state to another state will be random.
- The initial state is also random. It is given by the apriori probability of the initial state $p(y_0)\colon Y \to \mathbb{R}$.
- The state transition function state $\delta\colon Y \times V \to Y \times X$ generalizes to a stochastic transition function $\delta_s\colon Y \times V \to Y \times X$. The subscript at δ_s denotes "stochastic".
- This means that the corresponding state transition is random. The new state and the output is $(y',\, x) = \delta_s(y, v)$, where δ_s output is given by the conditional probability distribution function $p(y',\, x \,|\, y, v)$.

Autonomous stochastic finite automaton

- In this lecture, we deal with the special case of a stochastic finite automaton named the autonomous stochastic finite automaton.
- Its input alphabet V contains only one symbol. It expresses a special case, for which the automaton operation does not depend on input symbols.
- We will see later that the autonomous stochastic automaton is equivalent to (hidden) Markov chains.
- The state transition is analogical to the stochastic finite automaton with the exception that it does not depend on input symbols.
- Consider a set of states Y and a set of output symbols X. State transitions are governed by probabilities $p(y_0)$, $p(x_i, y_i \,|\, y_{i-1})$, $y_0 \in Y$, $y_i \in Y$, $x_i \in X$, $i = 1, 2, \ldots, n$.

Autonomous stochastic automaton in operation

The transitions between the hidden state y_i and the consecutive hidden state y_{i+1} is ruled by the probability distribution

$$p(\bar{x}, \bar{y}) = p(x_1, \ldots, x_n, y_0, \ldots, y_n) = p(y_0) \prod_{i=1}^{n} p(x_i, y_i | y_{i-1})$$

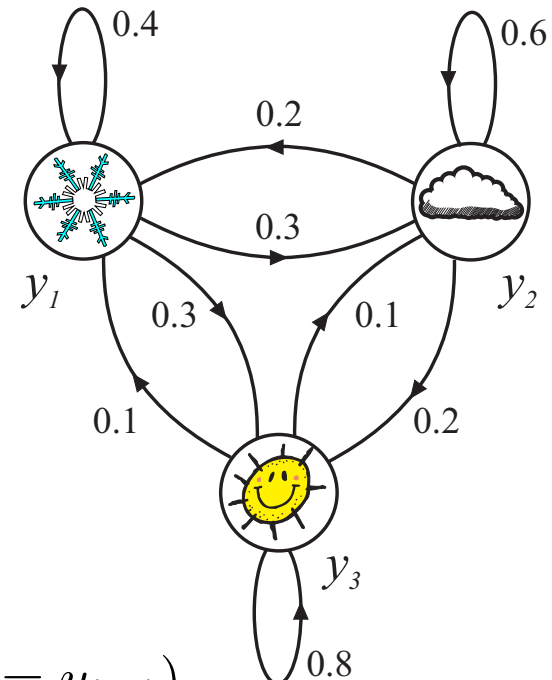
It means that the automaton:

- initially, generates a random state y_0 with the probability $p(y_0)$ and transfers to it;
- in the $i-$th step, it generates the tuple (x_i, y_i) with the probability $p(x_i, y_i|y_{i-1})$. It provides the symbol x_i at the output and transfers to the state y_i.

Example: Generative Markov weather model (1)

A stochastic finite automaton predicting the weather. No additional observations are considered.

- State $q = y_1$: precipitation (rain or snow). State $q = y_2$: clouds. State $q = y_3$: sunny.
- Transition matrix A between the q_t and the previous state q_{t-1} is independent of time.
  $a_{ij} = p(q_t = y_i | q_{t-1} = y_j),\ i, j \in \{1, 2, 3\}$



$$p(q_t = y_i) = A\, p(q_{t-1} = y_{i-1}) = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix} p(q_{t-1} = y_{i-1})$$

Courtesy: Ricardo Gutierrez-Osuna

Example: Generative Markov weather model (2)

- *Question 1*: Given that the weather on day $t = 1$ is sunny, what is the probability that the weather for the next seven days will be "sun, sun, rain, rain, sun, clouds, sun"?

  Answer 1:

$$
\begin{aligned}
&p(y_3, y_3, y_3, y_1, y_1, y_3, y_2, y_3 | \text{statistical model}) = \\
&\quad = p(y_3)\, p(y_3|y_3)\, p(y_3|y_3)\, p(y_1|y_3)\, p(y_1|y_1)\, p(y_3|y_1)\, p(y_2|y_3)\, p(y_3|y_2) = \\
&\quad = p(y_3)\, a_{33}\, a_{33}\, a_{13}\, a_{11}\, a_{31}\, a_{23}\, a_{32} = \\
&\quad = 1 \cdot 0.8 \cdot 0.8 \cdot 0.1 \cdot 0.4 \cdot 0.3 \cdot 0.1 \cdot 0.2 = 0.0001536
\end{aligned}
$$

- *Question 2*: What is the probability that the weather stays in the same known state y_i for exactly m consecutive days?

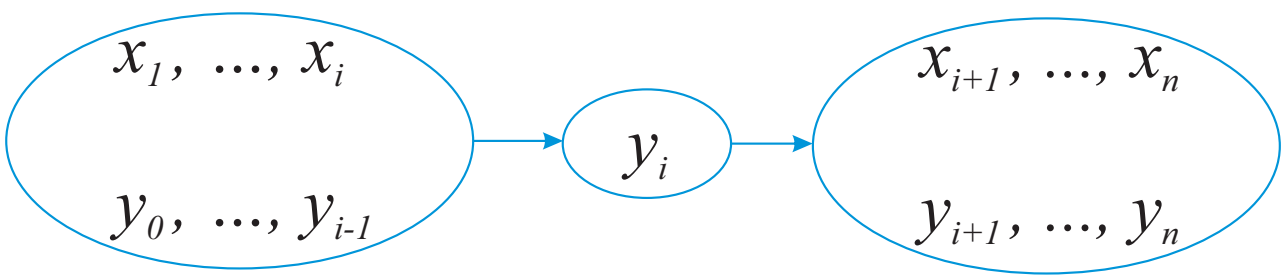
  Answer 2: $p(q_t = y_i, q_{t+1} = y_i \ldots q_{t+m} = y_{j \neq i}) = a_{ii}^{m-1}(1 - a_{ii})$

Markov sequences with hidden states

- Statistical model $p(\bar{x}, \bar{y}) = Y^n \times Y^{n+1} \to \mathbb{R}$.
- Markov chain / Markov condition:

  We assume that for all sequences $\bar{x} = (x_1^i, x_{i+1}^n)$ a $\bar{y} = (y_0^{i-1}, y_i, y_{i+1}^n)$ holds

$$p(\bar{x}, \bar{y}) = p(y_i)\; p(x_1^i, y_0^{i-1}|y_i)\; p(x_{i+1}^n, y_{i+1}^n|y_i)\;. \tag{1}$$



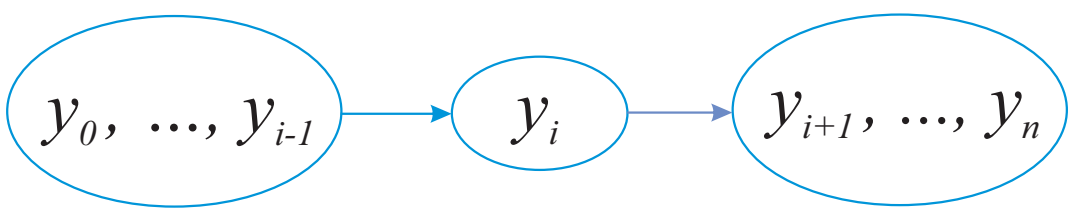
Markov condition for (hidden) states only

- We start from the Markov condition

$$p(\bar{x}, \bar{y}) = p(y_i)\; p(x_1^i, y_0^{i-1}|y_i)\; p(x_{i+1}^n, y_{i+1}^n|y_i)\;.$$

- For hidden states, after summing up (marginalization) over all possible observations $\bar{x}$, the Markov property holds for the sequence of hidden states $\bar{y}$

$$p(\bar{y}) = p(y_i)\; p(y_0^{i-1}|y_i)\; p(y_{i+1}^n|y_i)\;.$$



Hidden Markovian sequences (2)

We sum in the equation $p(\bar{x},\bar{y}) = p(y_i)\; p(x_1^i, y_0^{i-1}|y_i)\; p(x_{i+1}^n, y_{i+1}^n|y_i)$ along the sequence of hidden states y_{i+2}^n and after it along the sequence of observations x_{i+2}^n. We obtain

$$\begin{aligned} p(x_1^{i+1}, y_0^{i+1}) &= \sum_{x_{i+2}^n} \sum_{y_{i+2}^n} p(x,y) = p(x_1^i, y_0^i) \sum_{x_{i+2}^n} \sum_{y_{i+2}^n} p(x_{i+1}^n, y_{i+1}^n|y_i) \\ &= p(x_1^i, y_0^i)\; p(x_{i+1}, y_{i+1}|y_i) \end{aligned}$$

We use the previous expressions recursively and get

$$p(\bar{x},\bar{y}) = p(x_1,\ldots,x_n,y_0,\ldots,y_n) = p(y_0) \prod_{i=1}^{n} p(x_i, y_i|y_{i-1})$$

We simplified the enumeration of a complex function of $2\,n+1$ variables to the enumeration of n functions $p(x_i, y_i|y_{i-1})$ of three variables and one function $p(y_0)$ of one variable.

Interpretation of Markovian property

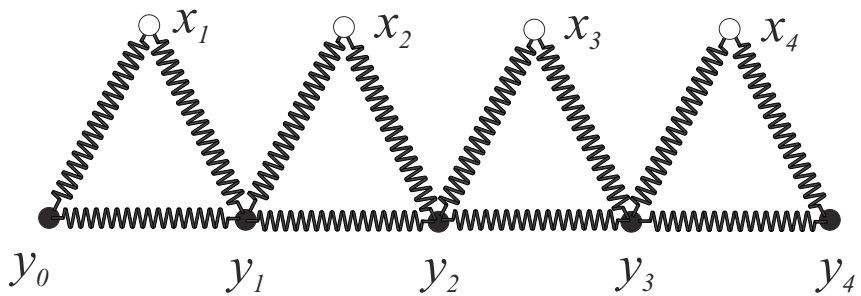
- Let consider every possible tuples of sequences $(x_1^n,\, y_0^n)$ fulfilling the Markov condition (1) from the slide 14.
- Let pick an arbitrary value i, $0 < i < n$. Let pick an arbitrary value of the hidden parameter $y_i = \sigma$.
- Let us pick an ensemble of sequences from possible tuples of Markov sequences $(x_1^n,\, y_0^n)$, for which $y_i = \sigma$ holds.
- Being a Markov sequence means consequently that parameters $(x_1, x_2, \ldots, x_i)$, $(y_0, y_1, \ldots, y_{i-1})$ in the selected ensemble of parameters are statistically independent on parameters $(x_{i+1},\, x_{i+2}, \ldots, x_n)$, $(y_{i+1},\, y_{i+2}, \ldots, y_n)$.

Beware of the incorrect interpretation

- The imprecisely simplified interpretation appears often: “Markov sequence is a such sequence, in which the future does not depend on its past but on the present only.”
- This interpretation is treacherous since while being incorrect it is very similar to the correct one.
- The mechanical model (analogy) of the Markov sequence provided on a next slide illustrates the intuition.

Mechanical model of the Markov sequence

- Let consider sequences x_1^4 a y_0^4 represented by vertices of a planar graph. Some vertices of the graph are connected by springs denoting statistical dependence in Markov statistical model.



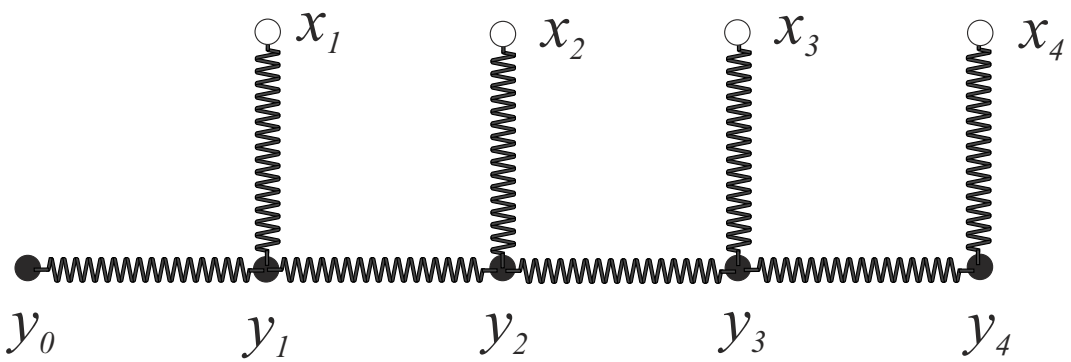
- Let imagine, for instance that, e.g. the vertex x_3 of a graph starts oscillating for a random reason. Due to direct and indirect mechanical bonds to other vertices of the graph, all (!) the graph vertices and not only y_2 and y_3 start oscillating too.
- When we fix vertices (values) $x_1, \ldots, x_4$, the values of vertices $y_0, \ldots, y_n$ are determined too.
- When we fix a vertex, e. g. the vertex y_3, the model decomposes into two independent parts: (a) vertices y_0, y_1, y_2, x_1, x_2, x_3; and (b) vertices x_4, y_4.

A special case: Decomposable statistical model

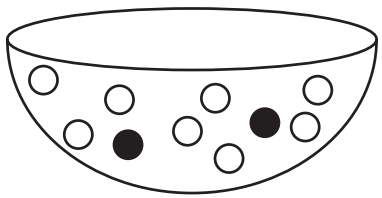
- It is a special case, which is considered often in the literature.
- It is assumed $p(x_i, y_i|y_{i-1}) = p(x_i|y_i)\, p(y_i|y_{i-1})$. In such a case, the following holds

$$p(\bar{x}, \bar{y}) = p(y_0) \prod_{i=1}^{n} p(x_i, y_i|y_{i-1}) = p(y_0) \prod_{i=1}^{n} p(x_i|y_i) \prod_{i=1}^{n} p(y_i|y_{i-1})$$

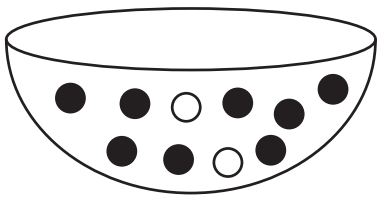
- The corresponding mechanical model looks like the following



Decomposable model example
Pulling out balls from urns



Urn 1



Urn 2

ball $x = \{\text{black, white}\}$
urn $y = \{1, 2\}$

$p(y = 1) = 0.5$
$p(y = 2) = 0.5$

$p(x = \text{white}|y = 1) = 0.8$
$p(x = \text{black}|y = 1) = 0.2$

$p(x = \text{white}|y = 2) = 0.2$
$p(x = \text{black}|y = 2) = 0.8$

Pulling out from urns alternatively

$p(y = 1|y = 2) = 1$
$p(y = 1|y = 1) = 0$

$p(y = 2|y = 2) = 0$
$p(y = 2|y = 1) = 1$

Three basic tasks with HMMs

1. Recognition also evaluation of a statistical model:

   (Called also the forward-backward task in literature. It is based on the dynamic programming). *Given:* parameters of HMM (of a statistical model). *The aim* is to calculate probabilities that we observed the sequence $\bar{x}$. The assigned class corresponds the most probable model. The approach is used for recognition (classification).

2. Seek for the most probable hidden states sequence:

   (Viterbi algorithm, dynamic programming). *Given:* A statistical model and a sequence of observations $\bar{x}$. *The aim* is finding the most probable sequence of hidden states $\bar{y}$.

3. Learning a statistical model from examples, aka parameter estimation of a Markov model:

   (Baum-Welsh re-estimation algorithm; explores EM algorithm). *Given:* a structure of a model, i.e., the number of hidden states and a training multi-sequence. *The aim* is find the parameters of a statistical model, i.e., probabilities $p(x_i, y_i|y_{i-1})$.

Recognition task; also enumeration of a particular statistical model

- Let a, b be two autonomous stochastic automata with the same number of states $|Y|$. The number of output symbols $|X|$ matches for both automata too.
- Let the statistical properties of the two automata a, b differ.

  The automaton a is described by $p_a(y_0)$ a $p_a(x_i, y_i \mid y_{i-1})$, $y_0 \in Y$, $y_i \in Y$, $x_i \in X$, $i = 1, 2, \ldots, n$.

  Similarly the automaton b is described by $p_b(y_0)$ a $p_b(x_i, y_i \mid y_{i-1})$.

  Note: The probabilities do not depend on the index i for simplicity. In general, the probabilities may depend on the index i. Our thoughts hold in this more general case too.
- The task evaluating the statistical model (also the recognition task) has to decide which of the automata generated the sequence of observations $x_1, x_2, \ldots, x_n$.

Recognition task (2)

- The recognition task can be expressed as the minimization of Bayesian risk, e.g., the number of erroneous decisions for simplicity. The formulation can extend to non-Bayesian tasks as, e.g., Neyman-Pearson tasks, minimax task.
- The appropriate marginal probabilities $p_a(x_1^n)$ and $p_b(x_1^n)$ have to be calculated for automata a, b.
- The decision is made, e.g., according to the maximal likelihood ratio $p_a(x_1^n)$ / $p_b(x_1^n)$.
  Recall the lecture on non-Bayesian pattern recognition, Neyman-Pearson task for dichotomic classification.
- The most difficult part of the task is calculating probabilities $p_a(x_1^n)$ and $p_b(x_1^n)$. The calculation is the same for automata a and b. Thus we do not show the index referring to a particular automaton.

Recognition algorithm

Recall Markov statistical model

$$p(\bar{x},\bar{y}) = p(x_1,\ldots,x_n,y_0,\ldots,y_n) = p(y_0)\prod_{i=1}^{n} p(x_i,y_i|y_{i-1})$$

We are interested in the marginal probability $p(\bar{x}) = \sum\limits_{y\in Y} p(\bar{x},\bar{y})$. It is expressed as a multiple sum

$$p(x) = \sum_{y} p(\bar{y},\bar{x}) = \sum_{y_0}\sum_{y_1}\cdots\sum_{y_{n-1}}\sum_{y_n} p(y_0)\prod_{i=1}^{n} p(x_i,y_i\,|\,y_{i-1})\,.$$

The direct calculation is unsuitable, because there are $|Y|^{n+1}$ summands. We will rearrange the formula by equivalent transformations into the usable form.

Recognition algorithm (2)

While summing up according to the state y_i, we factor out variables independent on y_i. Our starting point is the expression we are familar with already

$$p(\bar{x}) = \sum_k p(\bar{y}, \bar{x}) = \sum_{y_0} \sum_{y_1} \cdots \sum_{y_{n-1}} \sum_{y_n} p(y_0) \prod_{i=1}^{n} p(x_i, y_i \,|\, y_{i-1}) \,.$$

The formula is transformed to the following form after factoring out

$$\begin{aligned} p(\bar{x}) \quad = \quad & \sum_{y_0} p(y_0) \sum_{y_1} p(x_1, y_1 \,|\, y_0) \cdots \sum_{y_i} p(x_i, y_i \,|\, y_{i-1}) \\ \cdots \quad & \sum_{y_{n-1}} p(x_{n-1}, y_{n-1} \,|\, y_{n-2}) \sum_{k_n} p(x_n, y_n \,|\, y_{n-1}) \,. \end{aligned}$$

Recognition algorithm (3)

We aim at a recursive algorithm. After factoring out we get

$$\begin{aligned} p(\bar{x}) \quad &= \quad \sum_{y_0} p(y_0) \sum_{y_1} p(x_1, y_1 \,|\, y_0) \cdots \sum_{y_i} p(x_i, y_i \,|\, y_{i-1}) \\ &\cdots \quad \sum_{y_{n-1}} p(x_{n-1}, y_{n-1} \,|\, y_{n-2}) \sum_{y_n} p(x_n, y_n \,|\, y_{n-1}) \,. \end{aligned}$$

we mark partial sums for $i = 1, 2, \ldots, n$ using

$$\begin{aligned} f_i(y_{i-1}) \quad &= \quad \sum_{y_i} p(x_i, y_i \,|\, y_{i-1}) \sum_{y_{i+1}} p(x_{i+1}, y_{i+1} \,|\, y_i) \cdots \\ &\cdots \quad \sum_{y_{n-1}} p(x_{n-1}, y_{n-1} \,|\, y_{n-2}) \sum_{k_n} p(x_n, y_n \,|\, y_{n-1}) \end{aligned}$$

and become the algorithm.

Recognition algorithm (4)

The algorithm runs from the back to the beginning of the sequence

$$\left.\begin{aligned} f_n(y_{n-1}) &= \sum_{y_n} p(x_n, y_n \,|\, y_{n-1}); \\ f_i(y_{i-1}) &= \sum_{y_i} p(x_i, y_i \,|\, y_{i-1})\; f_{i+1}(y_i)\,, \quad i = 1, 2, \ldots, n-1\,; \\ p(\bar{x}) &= \sum_{y_0} p(y_0)\; f_1(y_0)\,. \end{aligned}\right\}$$

The number of operations is proportional to $|Y|^2\, n$.

The most probable sequence of hidden states

Task formulation

- The statistical model $p(\bar{x}, \bar{y}) = p(y_0) \prod_{i=1}^{n} p(x_i, y_i | y_{i-1})$ is given.
- We seek the decision strategy $q(\bar{x})$: $X^n \to Y^{n+1}$.
- Bayesian risk is $R(q(\bar{x})) = \sum_{\bar{x} \in X} \sum_{\bar{y} \in Y} p(\bar{x}, \bar{y})\, W(\bar{x}, q(\bar{x}))$.
- Let select a simple penalty function, for instance the number of wrong decisions

$$W(\bar{y}, q(\bar{x})) = \begin{cases} 0 & \text{pro } \bar{y} = q(\bar{x}) \,, \\ 1 & \text{pro } \bar{y} \neq q(\bar{x}) \,. \end{cases}$$

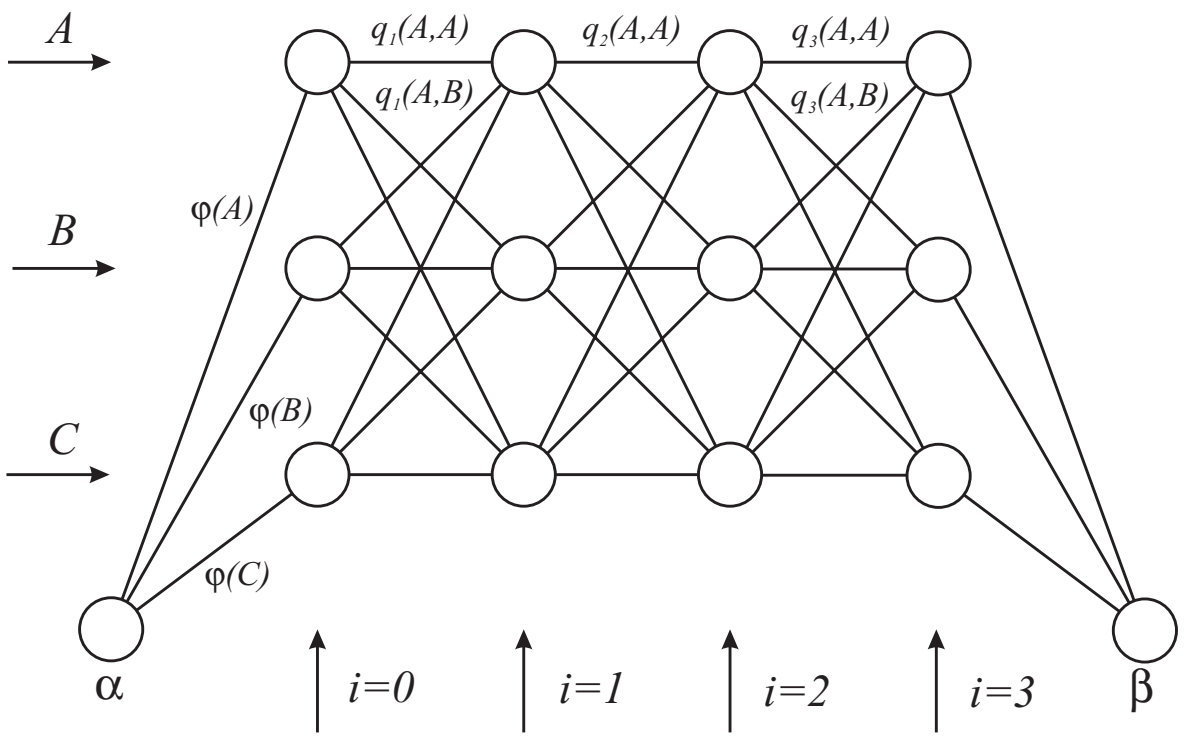
The aim is to find the strategy $q(\bar{x})$ minimizing the risk $R(q(\bar{x}))$.

Derivation of the Bayesian strategy

$$
\begin{aligned}
q(\bar{x}) &= \operatorname*{argmax}_{y \in Y^{n+1}} \frac{p(\bar{x}, \bar{y})}{\sum\limits_{y' \in Y^{n+1}} p(\bar{x}, \bar{y}')} = \operatorname*{argmax}_{y \in Y^{n+1}} p(\bar{x}, \bar{y}) \\
&= \arg\max_{y_0} \ldots \max_{y_n} p(y_0) \prod_{i=1}^{n} p(x_i, y_i | y_{i-1}) \\
&= \arg\max_{y_0} \ldots \max_{y_n} \log \left(p(y_0) \prod_{i=1}^{n} p(x_i, y_i | y_{i-1}) \right) \\
&= \arg\max_{y_0} \ldots \max_{y_n} \left(\underbrace{\log p(y_0)}_{\varphi(y_0)} + \sum_{i=1}^{n} \underbrace{\log p(x_i, y_i | y_{i-1})}_{q_i(y_{i-1}, y_i)} \right)
\end{aligned}
$$

Formulation as the seek for the shortest path in a graph

- A special oriented graph (trellis) with vertices and edges ordered left to right. The initial vertex is α and the goal vertex is β. The remaining $|Y|(n+1)$ intermediate vertices are indexed by a tuple (σ, i), $\sigma \in Y$, $i = 0, 1, \ldots, n$.
- Example of a graph (trellis) for $n = 3$ and hidden states $Y = \{A, B, C\}$.



The shortest path algorithm

Dynamic programming. The graph has a special form that ensures ordering. The algorithm analogy – messengers.

$f_i(\sigma)$ is the length of the shortest path ($\sim$ time) from vertex α to vertex (σ, i).

Algoritmus A. Viterbi 1967 (independently T. Vincjuk 1968):

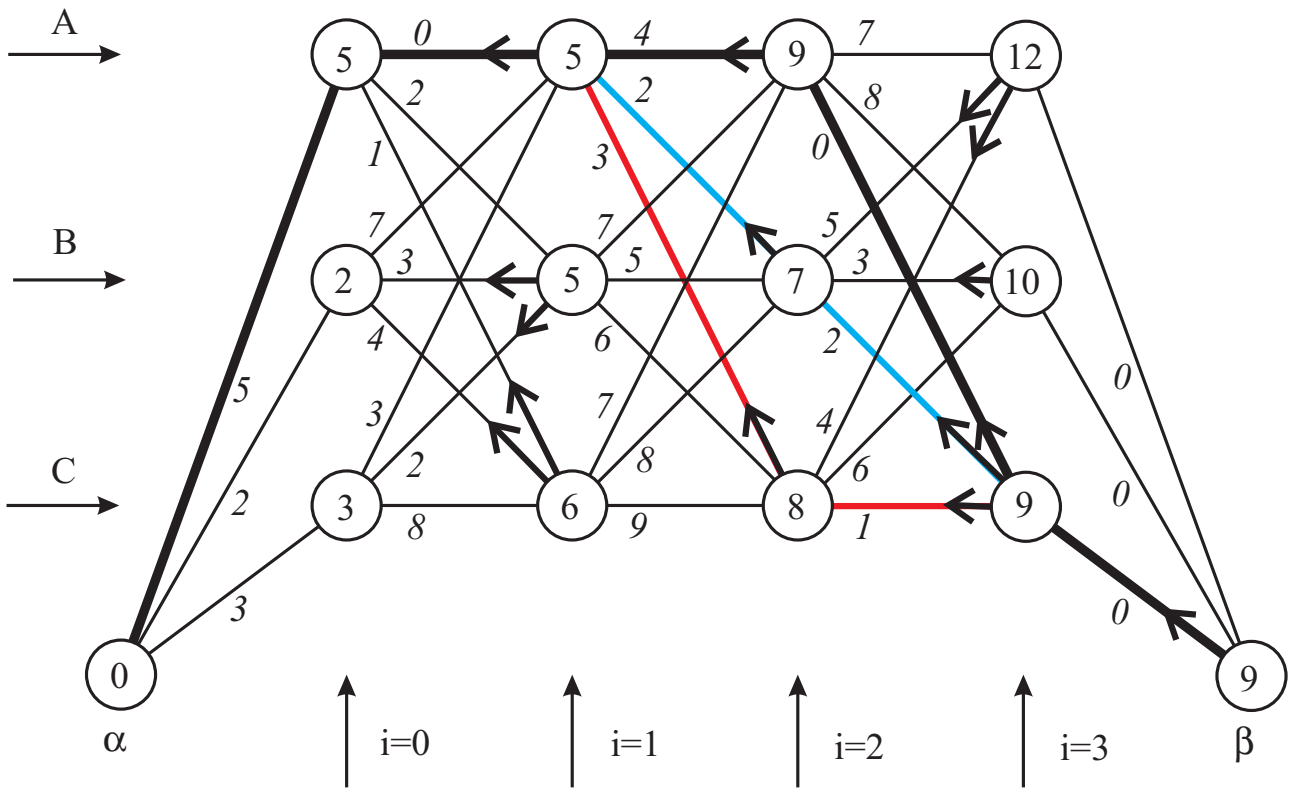
- $f_0(\sigma) = \varphi(\sigma)$
- Repeatedly for $i = 1, \ldots, n, \quad \sigma \in Y$

  $f_i(\sigma) = \min\limits_{\sigma' \in Y} (f_i(\sigma') + q_i(\sigma', \sigma))$. The path of the messenger, who came to the vertex first.

  $\mathrm{ind}_i(\sigma) = \operatorname{argmin}\limits_{\sigma' \in y} (f_i(\sigma') + q_i(\sigma', \sigma))$. The vertex, from which the first messenger came.
- Finally:

  $y_n = \operatorname{argmin}\limits_{\sigma \in Y} f_n(\sigma), \quad y_{i-1} = \mathrm{ind}_i(y_i)$. Reconstruction of the shortest path.

Viterbi algorithm example



Arrows denote $\mathrm{ind}_i(\sigma)$. Arrows are used while seeking the shortest path. There can be than one shortes path. In our example, these are besides **AAAC** also AABC or AACC.