International Journal of Pure and Applied Mathematics
Volume 115 No. 3 2017, 621-639
ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)
url: http://www.ijpam.eu
doi: 10.12732/ijpam.v115i3.15

INFERENCE AND LEARNING IN STOCHASTIC AUTOMATA

Karl-Heinz Zimmermann

Department of Electrical Engineering,
Computer Science, Mathematics
Hamburg University of Technology
21071, Hamburg, GERMANY

Abstract: Machine learning provides algorithms that can learn from data and make inferences or predictions on data. Stochastic automata are a class of input/output devices which can model components in machine learning scenarios. In this paper, we provide an inference algorithm for stochastic automata which is related to the Viterbi algorithm. Moreover, we specify a learning algorithm using the expectation-maximization technique and describe a more efficient implementation which is related to the Baum-Welch algorithm.

AMS Subject Classification: 68Q70, 68T05, 16Y60
Key Words: stochastic automaton, tropical semiring, dynamic programming, inference, learning

1. Introduction

The theory of discrete stochastic systems has been initiated by the work of Shannon [16] and von Neumann [6]. While Shannon has considered memoryless communication channels and their generalization by introducing states, von Neumann has studied the synthesis of reliable systems from unreliable

Received: July 14, 2017
Revised: June 3, 2017
Published: July 27, 2017

components. The fundamental work of Rabin and Scott [11] about deterministic finite-state automata has led to two generalizations. First, the generalization of transition functions to conditional distributions studied by Carlyle [7] and Starke [17]. This in turn provides a generalization of discrete-time Markov chains in which the chains are governed by more than one transition probability matrix. Second, the generalization of regular sets by introducing stochastic acceptors as described by Rabin [10].

The hidden Markov model (HMM) is a Bayesian network which is related to stochastic automata [3, 13, 20]. It models a Markov chain with observed (emission) data and unobserved (hidden) states. The inference problem in a HMM is to find the most probable state transitions given an emission sequence. The famous Viterbi algorithm is a dynamic programming algorithm which solves this problem [19]. The learning of parameters in a HMM is the task to find the best set of state transitions and emission probabilities given a set of emission sequences [9]. Tractable algorithms for solving this problem are not known, but local maximum likelihood algorithms such as the Baum-Welch algorithm as a special case of the famous expectation-maximization algorithm can be efficiently applied [8, 20].

Stochastic automata are used in spoken language understanding for the recognition and interpretation of speech signals. The most successful parsing algorithm for finding the most likely sequence of spoken words is Viterbi decoding along with beam search [14]. A unified formalism for building a wide class of language models are the variable multi-gram stochastic automata [12].

Stochastic automata can be extended to the modeling of soft real-time systems by using methods from timed automata and generalized semi-Markov chains. Such automata allow to model constraints like "the system should perform an action before time t in 90% of the cases" [1, 2].

Stochastic automata can be used as building blocks in situations of machine learning where detailed mathematical description is missing and feature management is noisy. The arrangement of stochastic automata in the form of teams or hierarchies could lead to solutions of complex learning problems [18].

In this paper, we provide an inference algorithm for stochastic automata which is related to the Viterbi algorithm. Moreover, we specify a learning algorithm using the expectation-maximization technique and describe a variant of the Baum-Welch algorithm. The text is to a large extent self-contained and also suitable to non-experts in this field.

2. Mathematical Preliminaries

A *stochastic automaton* (SA) [5, 15] is a quadruple $A = (S, \Sigma, \Omega, p)$, where S is a nonempty finite set of *states*, Σ is an alphabet of *input symbols*, Ω is an alphabet of *output symbols*, and for each $s \in S$ and $a \in \Sigma$, $p(\cdot, \cdot \mid a, s)$ is a conditional probability distribution on $\Omega \times S$.

Note that a conditional probability distribution $p(\cdot, \cdot \mid a, s)$ on $\Omega \times S$ consists of nonnegative numbers $p(b, s' \mid a, s)$ for all $s' \in S$ and $b \in \Omega$ such that

$$\sum_{b \in \Omega} \sum_{s' \in S} p(b, s' \mid a, s) = 1, \quad a \in \Sigma, s \in S. \tag{1}$$

Given a conditional probability distribution $p(\cdot, \cdot \mid a, s)$ on $\Omega \times S$, we define a probability distribution $\hat{p}$ on $\Omega^* \times S$ recursively as follows.

- For each $s, s' \in S$,

$$\hat{p}(\epsilon, s' \mid \epsilon, s) = \begin{cases} 1 & \text{if } s = s', \\ 0 & \text{if } s \neq s', \end{cases} \tag{2}$$

  where ϵ denote the empty word in Σ^* and Ω^*.

- For all $s, s' \in S$, $x \in \Sigma^*$, and $y \in \Omega^*$ with $|x| \neq |y|$,

$$\hat{p}(y, s' \mid x, s) = 0. \tag{3}$$

- For all $s, s' \in S$, $a \in \Sigma$, $x \in \Sigma^*$, $b \in \Omega$, and $y \in \Omega^*$,

$$\hat{p}(yb, s' \mid xa, s) = \sum_{t \in S} \hat{p}(y, t \mid x, s) \cdot p(b, s' \mid a, t). \tag{4}$$

Then $\hat{p}(\cdot, \cdot \mid x, s)$ is a conditional probability distribution on $\Omega^* \times S$ and so we have

$$\sum_{y \in \Omega^*} \sum_{s' \in S} p(y, s' \mid x, s) = 1, \quad x \in \Sigma^*, s \in S. \tag{5}$$

A stochastic automaton works serially and synchronously. It reads an input word symbol by symbol and after reading an input symbol it emits an output symbol and transits into another state. In particular, if the automaton starts in state s and reads the word x, then with probability $\hat{p}(y, s' \mid x, s)$ it will end in state s' emitting the word y and taking all intermediate states into account.

Note that the measures p and $\hat{p}$ coincide on the set $\Omega \times S \times \Sigma \times S$ if we put $x = y = \epsilon$ in (4). Therefore, we write p instead of $\hat{p}$.

Proposition 2.1. *For all $x, x' \in \Sigma^*$, $y, y' \in \Omega^*$, and $s, s' \in S$ with $|x| = |y|$,*

$$p(yy', s' \mid xx', s) = \sum_{t \in S} p(y, t \mid x, s) \cdot p(y', s' \mid x', t). \tag{6}$$

The behavior of a stochastic automaton can be described by probability matrices. To see this, let A be a stochastic automaton with state set $S = \{s_1, \ldots, s_n\}$. For each pair of input and output symbols $a \in \Sigma$ and $b \in \Omega$, put

$$p_{ij}(b \mid a) = p(b, s_j \mid a, s_i), \quad 1 \leq i, j \leq n, \tag{7}$$

and define the real-valued $n \times n$ matrix

$$P(b \mid a) = (p_{ij}(b \mid a))_{1 \leq i,j \leq n}. \tag{8}$$

Note that the matrix $P(b \mid a)$ is *substochastic*, i.e., it is a square matrix with nonnegative entries and by (5) each row adds up to at most 1. The elements of $P(b \mid a)$ provide the transition probabilities between the states if the symbol a is read and the symbol b is emitted. This definition can be extended to strings of input and output symbols. For this, note that by (2) we have

$$P(\epsilon \mid \epsilon) = I_n, \tag{9}$$

where I_n is the $n \times n$ unit matrix. Moreover, if $x \in \Sigma^*$ and $y \in \Omega^*$ with $|x| \neq |y|$, then by (3) we have

$$P(x \mid y) = O_n, \tag{10}$$

where O_n is the $n \times n$ zero matrix. Furthermore, if $a \in \Sigma$, $x \in \Sigma^*$, $b \in \Omega$, and $y \in \Omega^*$, then by (4) we have

$$P(yb \mid xa) = P(y \mid x) \cdot P(b \mid a). \tag{11}$$

By Prop. 2.1 and the associativity of matrix multiplication, we obtain the following.

Proposition 2.2. *For all $x, x' \in \Sigma^*$ and $y, y' \in \Omega^*$ with $|x| = |y|$,*

$$P(yy' \mid xx') = P(y \mid x) \cdot P(y' \mid x'). \tag{12}$$

It follows by induction that if $x = x_1 \ldots x_k \in \Sigma^*$ and $y = y_1 \ldots y_k \in \Omega^*$, then

$$P(y \mid x) = P(y_1 \mid x_1) \cdots P(y_k \mid x_k). \tag{13}$$

Example 2.3. Consider the stochastic automaton $A = (\{a\}, \{b\}, \{s_1, s_2\}, p)$ with conditional probabilities (Fig. 1)

$$p(b, s_1 \mid a, s_1) = \frac{2}{3}, \quad p(b, s_2 \mid a, s_1) = \frac{1}{3}, \quad \text{and} \quad p(b, s_2 \mid a, s_2) = 1.$$

The corresponding (substochastic) matrix is

$$P(a) = P(b \mid a) = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \\ 0 & 1 \end{pmatrix}.$$

Thus for each integer $k \geq 1$,

$$P(a^k) = \begin{pmatrix} \frac{2^k}{3^k} & \frac{3^k - 2^k}{3^k} \\ 0 & 1 \end{pmatrix}.$$

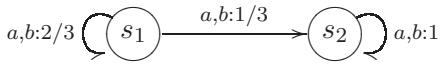

Figure 1: State diagram of A.

For each input word $x \in \Sigma^*$, the stochastic matrix $P(x)$ can be viewed as generating a discrete-time Markov chain. Thus the behavior of a stochastic automaton is an interleaving of Markov chains each of which corresponding to a single input symbol.

The probabilistic inference of the emission marginals can be used to find the most probable state sequences which produce a given emission sequence. For this, we introduce the tropical semiring.

A semiring is an algebraic structure similar to a ring, but without the requirement that each element must have an additive inverse. A prominent example of a semiring is the so-called tropical semiring. More specifically, a *semiring* is a non-empty set R together with two binary operations, addition $+$ and multiplication $\cdot$, such that $(R, +)$ is a commutative monoid with identity element 0, $(R, \cdot)$ is a monoid with identity element 1, the multiplication distributes over addition, i.e., for all $a, b, c \in R$,

$$a \cdot (b + c) = (a \cdot b) + (a \cdot c) \quad \text{and} \quad (a + b) \cdot c = (a \cdot c) + (b \cdot c), \tag{14}$$

and the multiplication with 0 annihilates R, i.e., for all $a \in R$, $a \cdot 0 = 0 = 0 \cdot a$.

A *commutative* semiring is a semiring whose multiplication is commutative. An *idempotent* semiring is a semiring whose addition is idempotent, i.e., for all $a \in R$, $a + a = a$.

Each ring is also a semiring. The set of natural numbers $\mathbb{N}_0$ forms a commutative semiring with the ordinary addition and multiplication. Likewise, the set of non-negative real numbers forms commutative semirings.

The set $\mathbb{R} \cup \{\infty\}$ together with the operations

$$x \oplus y = \min\{x, y\} \quad \text{and} \quad x \odot y = x + y, \quad x, y \in \mathbb{R} \cup \{\infty\}, \tag{15}$$

with $x+\infty = \infty$ for all $x \in \mathbb{R}\cup\{\infty\}$ forms an idempotent commutative semiring with additive identity ∞ and multiplicative identity 0. Note that additive and multiplicative inverses may not exist. For instance, the equations $1 \oplus x = 2$ and $\infty \odot x = 1$ have no solutions $x \in \mathbb{R} \cup \{\infty\}$. This semiring is also known min-plus semiring or *tropical semiring*. The attribute "tropical" was coined by French scholars (1998) in honor of the Brazilian mathematician Imre Simon who studied the tropical semiring in the early 1960s.

Proposition 2.4. *The mapping* $\phi : \mathbb{R}_{\geq 0} \to \mathbb{R} \cup \{\infty\} : x \mapsto -\log x$ *is bijective and monotonically decreasing with* $\phi(0) = \infty$, $\phi(1) = 0$, *and*

$$\phi(x \cdot y) = \phi(x) \odot \phi(y), \quad x, y \in \mathbb{R}_{\geq 0}. \tag{16}$$

The mapping ϕ is called the *tropicalization* of the ordinary semiring $(\mathbb{R}_{\geq 0}, +, \cdot)$. In this way, large probabilities are mapped to small weights and vice versa.

3. Probabilistic Inference of I/O Marginals

Stochastic automata are abstract machines with an input/output (I/O) interface. Suppose an external observer can see both, an input string and the output string responded by the machine. Then the problem is to find the most probable state sequences which correpond to this I/O communication.

Let $A = (S, \Sigma, \Omega, p)$ be a stochastic automaton with l-element state set S, l_1-element input set Σ, and l_2-element output set Ω. A stochastic automaton is a specific belief network. To define the network, let $n \geq 1$ be an integer. Let $X_1, \ldots, X_n$ be random variables with common finite state set Σ, let $S_1, \ldots, S_{n+1}$ be random variables with common finite state set S, and let $Y_1, \ldots, Y_n$ be random variables with common finite state set Ω. The belief network is shown in Fig. 2. Thus the joint probability distribution is given by

$$p_{X,S,Y} = p_{X_1} p_{S_1} p_{Y_1,S_2|X_1,S_1} p_{Y_2,S_3|X_2,S_2} \cdots p_{Y_n,S_{n+1}|X_n,S_n}. \tag{17}$$

We assume for simplicity that the initial distributions p_{X_1} and p_{S_1} are uniform; i.e., $p_{X_1}(x) = \frac{1}{l_1}$ for all $x \in \Sigma$ and $p_{S_1}(s) = \frac{1}{l}$ for all $s \in S$. Furthermore,

the network is assumed to be *homogeneous* in the sense that the conditional distributions $p_{Y_i,S_{i+1}|X_i,S_i}$ are independent of the index i, $1 \leq i \leq n$. Therefore, we put

$$\theta_{b,s';a,s} = p_{Y_i,S_{i+1}|X_i,S_i}(b, s' \mid a, s), \quad s, s' \in S,\ a \in \Sigma,\ b \in \Omega,\ 1 \leq i \leq n. \tag{18}$$

It follows that the joint probability distribution has the form

$$p_{X,S,Y}(x_1, \ldots, x_n, s_1, \ldots, s_{n+1}, y_1, \ldots, y_n) = \frac{1}{l \cdot l_1} \theta_{y_1,s_2;x_1,s_1} \cdots \theta_{y_n,s_{n+1};x_n,s_n}. \tag{19}$$

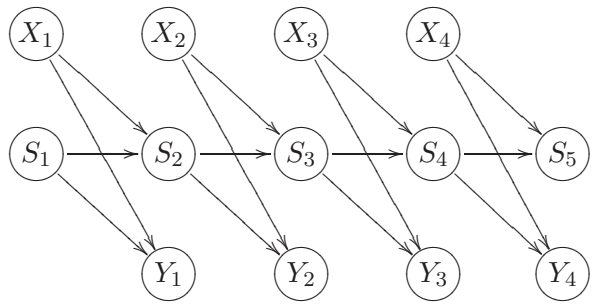

Figure 2: Belief network of stochastic transducer with $n = 4$.

The probability of input sequence $x = x_1 \ldots x_n \in \Sigma^n$ and output sequence $y = y_1 \ldots x_y \in \Omega^n$ is given by the marginal distribution

$$p_{X,Y}(x, y) = \sum_{s \in S^{n+1}} p_{X,S,Y}(x, s, y). \tag{20}$$

The corresponding sum-product decomposition yields

$$p_{X,Y}(x, y) = \frac{1}{l \cdot l_1} \sum_{s_{n+1} \in S} \left(\sum_{s_n \in S} \theta_{y_n,s_{n+1};x_n,s_n} \left(\cdots \left(\sum_{s_1 \in S} \theta_{y_1,s_2;x_1,s_1} \right) \cdots \right) \right). \tag{21}$$

This decomposition can be exploited to evaluate the probability $p(x, y)$ by using an $n \times l$ table M:

$$M[0, s] := \frac{1}{l \cdot l_1}, \quad s \in S,$$

$$\begin{aligned} M[k,s] &:= \sum_{s' \in S} \left(\theta_{y_k,s;x_k,s'} \cdot M[k-1,s']\right), \quad s \in S,\ 1 \le k \le n, \qquad (22)\\ w(x,y) &:= \sum_{s \in S} M[n,s]. \end{aligned}$$

The time complexity of this algorithm is $O(l^2 n)$, since the table M has size $O(ln)$ and each table entry is computed in $O(l)$ steps.

Given the sequences $x \in \Sigma^n$ and $y \in \Omega^n$, the objective is to find one (or all) state sequences $s \in S^{n+1}$ with maximum likelihood

$$p_{S|X,Y}(s \mid x,y) = \frac{p_{X,S,Y}(x,s,y)}{p_{X,Y}(x,y)}. \qquad (23)$$

Since the observed sequence pair (x,y) is fixed, the likelihood $p_{S|X,Y}(s \mid x,y)$ is directly proportional to the joint probability $p_{X,S,Y}(x,s,y)$ provided that $p_{X,Y}(x,y) > 0$. Suppose that $p_{X,Y}(x,y) > 0$. Then the aim is to find the state sequences $\bar{s} \in S^{n+1}$ with the property

$$\bar{s} \;=\; \mathrm{argmax}_{s \in S^{n+1}}\{p_{X,S,Y}(x,s,y)\}. \qquad (24)$$

Each optimal state sequence $\bar{s}$ is called an *explanation* of the I/O sequence pair (x,y). The explanations can be found by tropicalization. For this, put $w(x,y) = -\log p_{X,Y}(x,y)$ and $w(x,s,y) = -\log p_{X,S,Y}(x,s,y)$ for all $x \in \Sigma^n$, $y \in \Omega^n$, and $s \in S^{n+1}$. Then the tropicalization yields

$$w(x,y) = \bigoplus_{s \in S^{n+1}} w(x,s,y). \qquad (25)$$

The explanations $\bar{s}$ are obtained by evaluation in the tropical semiring,

$$\bar{s} \;=\; \mathrm{argmin}_{s \in S^{n+1}}\{w(x,s,y)\}. \qquad (26)$$

The value $w(x,y)$ can be efficiently computed by tropicalizing the sum-product decomposition of the marginal probability $p_{X,Y}(x,y)$. For this, we put $u_{b,s';a,s} = -\log \theta_{y,s';x,s}$ for $a \in \Sigma$, $b \in \Omega$, and $s, s' \in S$. By replacing the sums by tropical sums and the products by tropical products in the sum-product decomposition (21), we obtain (up to a constant)

$$w(x,y) = \qquad (27)$$
$$\bigoplus_{s_{n+1} \in S} \left(\bigoplus_{s_n \in S} u_{y_n,s_{n+1};x_n,s_n} \odot \left(\cdots \odot \left(\bigoplus_{s_1 \in S} u_{y_1,s_2;x_1,s_1} \right) \cdots \right) \right).$$

This yields the following result.

Proposition 3.1. *Let $x \in \Sigma^n$ and $y \in \Omega^n$. The tropicalization $w(x,y)$ of the marginal probability $p_{X,Y}(x,y)$ provides the explanations of the I/O sequence pair (x,y).*

The tropicalized term $w(x,y)$ can be computed by evaluating iteratively the tropicalized sum-product decomposition (27) by using an $n \times l$ table M:

$$\begin{aligned} M[0,s] &:= 0, \quad s \in S, \\ M[k,s] &:= \bigoplus_{s' \in S} \left(u_{y_k,s;x_k,s'} \odot M[k-1,s'] \right), \quad s \in S,\ 1 \le k \le n, \\ w(x,y) &:= \bigoplus_{s \in S} M[n,s]. \end{aligned} \tag{28}$$

This algorithm is a variant of the Viterbi algorithm. It consists of a forward algorithm evaluating the data as given by Alg. 1 and a backward algorithm which provides one (or all) optimal state sequences. The latter is achieved by recording in each step one (or all) states which attain the minimum in the minimization step. This information can already be recorded by the forward algorithm. Then the trace back of all optimal decisions (states) from the last to the first position can provide one (or all) explanations. The time complexity of the algorithm is $O(l^2 n)$, since the table M has size $O(ln)$ and each table entry is computed in $O(l)$ steps.

Example 3.2. The arbitrarily varying channel (AVC) is a more realistic network channel than the more idealized binary symmetric channel [4]. An AVC has an input alphabet Σ, an output alphabet Ω, and a state set S. During transmission of an input symbol, the state of the set S can vary arbitrarily at each time step. The conditional probabilities of an AVC can be defined as

$$p(b, s' \mid a, s) = p'(b \mid a, s) \cdot p''(s' \mid s).$$

where $p'(b \mid a, s)$ is the conditional probability of receiving the symbol b when the symbol a has been transmitted in state s, and $p''(s' \mid s)$ is the conditional probability of moving from state s to state s'. Consider the arbitrarily varying channel (AVC) given by the stochastic automaton $A = (S, \Sigma, \Omega, p)$ with state set $S = \{s_1, s_2\}$, input alphabet $\Sigma = \{0,1\}$, output alphabet $\Omega = \{0,1\}$, and conditional probabilities

p'	$0,s_1$	$0,s_2$	$1,s_1$	$1,s_2$
0	0.50	0.30	0.40	0.60
1	0.50	0.70	0.60	0.40

and

p''	s_1	s_2
s_1	0.70	0.50
s_2	0.30	0.50

Algorithm 1 Forward algorithm.

Require: Sequences $x \in \Sigma^n$, $y \in \Omega^n$, scores $(u_{b,s';a,s})$
Ensure: Tropicalized term $w(x, y)$
  $M \leftarrow \text{matrix}[0..n, 1..l]$
  for $s \leftarrow 1$ to l **do**
    $M[0, s] \leftarrow 0$
  end for
  for $k \leftarrow 1$ to n **do**
    for $s \leftarrow 1$ to l **do**
      $M[k, s] \leftarrow \infty$
      for $s' \leftarrow 1$ to l **do**
        $M[k, s] \leftarrow \min\{M[k, s], u_{y_k,s;x_k,s'} + M[k-1, s']\}$ {Record all s which attain the minimum}
      end for
    end for
  end for
  $w \leftarrow \infty$
  for $s \leftarrow 1$ to l **do**
    $w \leftarrow \min\{w, M[n, s]\}$ {Record all s which attain the minimum}
  end for
  return $w = w(x, y)$

Then we have

$$\begin{array}{lll} p(0,s_1|0,s_1) = 0.35, & p(0,s_2|0,s_1) = 0.15, & p(1,s_1|0,s_1) = 0.35, \\ p(1,s_2|0,s_1) = 0.15, & p(0,s_1|0,s_2) = 0.15, & p(0,s_2|0,s_2) = 0.15, \\ p(1,s_1|0,s_2) = 0.35, & p(1,s_2|0,s_2) = 0.35, & p(0,s_1|1,s_1) = 0.28, \\ p(0,s_2|1,s_1) = 0.12, & p(1,s_1|1,s_1) = 0.42, & p(1,s_2|1,s_1) = 0.18, \\ p(0,s_1|1,s_2) = 0.30, & p(0,s_2|1,s_2) = 0.30, & p(1,s_1|1,s_2) = 0.20, \\ p(1,s_2|1,s_2) = 0.20. & & \end{array}$$

The corresponding tropicalized values are

$$\begin{array}{lll} u_{0,s_1|0,s_1} = 1.05, & u_{0,s_2|0,s_1} = 1.90, & u_{1,s_1|0,s_1} = 1.05, \\ u_{1,s_2|0,s_1} = 1.90, & u_{0,s_1|0,s_2} = 1.90, & u_{0,s_2|0,s_2} = 1.90, \\ u_{1,s_1|0,s_2} = 1.05, & u_{1,s_2|0,s_2} = 1.05, & u_{0,s_1|1,s_1} = 1.27, \\ u_{0,s_2|1,s_1} = 2.12, & u_{1,s_1|1,s_1} = 0.87, & u_{1,s_2|1,s_1} = 1.71, \\ u_{0,s_1|1,s_2} = 1.20, & u_{0,s_2|1,s_2} = 1.20, & u_{1,s_1|1,s_2} = 1.61, \\ u_{1,s_2|1,s_2} = 1.61. & & \end{array}$$

Consider the input sequence $x = 1000$ and the output sequence $y = 0101$. The calculation of the forward algorithm is given by the trellis in Fig. 3. The solid arrows show where the minima are attained. The trace back given by the paths of solid arrows provides two explanations: $s_1 s_1 s_1 s_1$ and $s_2 s_2 s_1 s_1$ (with minimal score 4.35).

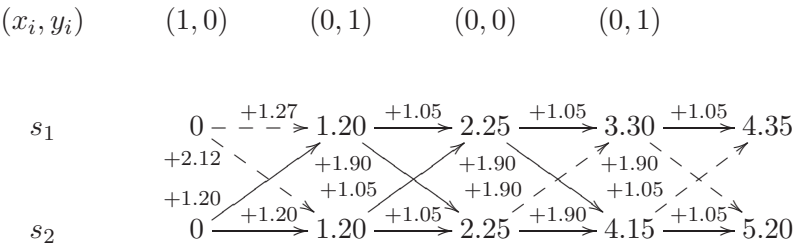

Figure 3: Trellis for the I/0 sequences $x = 1000$ and $y = 0101$.

4. Learning the Conditional Probability Potentials

The aim is to learn or estimate the conditional probabilities of a stochastic automaton by using sample data. The estimation can be achieved by the maximum likelihood method. For this, let $A = (S, \Sigma, \Omega, p)$ be a stochastic automaton with $l = |S|$, $l_1 = |\Sigma|$, and $l_2 = |\Omega|$, and let $n \geq 1$. Take the parameter set

$$\Theta = \{\theta = (\theta_{b,s';a,s}) \mid \theta_{b,s';a,s} \geq 0, \sum_{b,s'} \theta_{b,s';a,s} = 1\}. \tag{29}$$

where

$$\theta_{b,s';a,s} = p(b, s' \mid a, s), \quad a \in \Sigma,\ s, s' \in S,\ b \in \Omega. \tag{30}$$

The aim is to estimate these probabilities by making use of a sample set. For this, assume that there is a collection $D = (d_1, \ldots, d_N)$ of N independent samples called *database*, where $d_r = (x_r, s_r, y_r) \in \Sigma^n \times S^{n+1} \times \Omega^n$ denotes the r-th sample, $1 \leq r \leq N$. Then the joint probability of the sample d_r depending on the parameters is given by

$$p_{X,S,Y|\Theta}(d_r \mid \theta) = \frac{1}{l \cdot l_1} \prod_{i=1}^{n} \theta_{y_{r,i}, s_{r,i+1}; x_{r,i}, s_{r,i}}. \tag{31}$$

Thus the likelihood function $L = L_{X,S,Y}$ is given by

$$L(\theta) \;=\; \prod_{r=1}^{N} p_{X,S,Y|\Theta}(d_r \mid \theta) = \prod_{(x,s,y)} p_{X,S,Y|\Theta}(x,s,y \mid \theta)^{u_{x,s,y}}, \tag{32}$$

where $u_{x,s,y}$ is the number of times the sequence (x,s,y) is observed in the sample set. Therefore, we have

$$\sum_{(x,s,y)} u_{x,s,y} = N. \tag{33}$$

Let $v_{b,s';a,s}$ be the number of times the parameter $\theta_{b,s';a,s}$ occurs in the likelihood function $L(\theta)$. Then the likelihood function can be written (up to a constant) as

$$L(\theta) = \prod_{a\in\Sigma} \prod_{s,s'\in S} \prod_{b\in\Omega} \theta_{b,s';a,s}^{v_{b,s';a,s}}. \tag{34}$$

The corresponding log-likelihood function $\ell = \ell_{X,S,Y}$ is

$$\ell(\theta) = \log L(\theta) = \sum_{a\in\Sigma} \sum_{s,s'\in S} \sum_{b\in\Omega} v_{b,s';a,s}\theta_{b,s';a,s}. \tag{35}$$

The data $v = (v_{b,s';y,s})$ form the sufficient statistic of the model. They can be obtained from the given data $u = (u_{x,s,y})$ by the linear transformation

$$v = A \cdot u, \tag{36}$$

where A is an integral matrix with $d = l^2 l_1 l_2$ rows labeled by the quadruples $(b,s';a,s)$ with $a \in \Sigma$, $s,s' \in S$, and $b \in \Omega$. Moreover, the matrix has $m = l_1^n l^{n+1} l_2^n$ columns labeled by the triples $(x,s,y) \in \Sigma^n \times S^{n+1} \times \Omega^n$. The matrix has entry k in row $(b,s';a,s)$ and column (x,s,y) if the parameter $\theta_{b,s';a,s}$ occurs k times in $p(x,s,y)$. Note that the matrix has column sum n, since the quantity $p(x,s,y)$ has n factors.

Example 4.1. Consider the stochastic automaton $A = \{\{0,1\},\{a\},\{b\},p\}$ and let $n = 2$. The associated 4×8 matrix is as follows,

	$aa,000,bb$	$aa,001,bb$	$aa,010,bb$	$aa,011,bb$	$aa,100,bb$	$aa,101,bb$	$aa,110,bb$	$aa,111,bb$
$b,0;0,a$	2	1	0	0	1	0	0	0
$b,0;1,a$	0	1	1	1	0	1	0	0
$b,1;0,a$	0	0	1	0	1	1	1	0
$b,1;1,a$	0	0	0	1	0	0	1	2

Proposition 4.2. *The maximum likelihood estimate of the likelihood function $L(\theta)$ is given by*

$$\hat{\theta}_{b,s';a,s} = \frac{v_{b,s';a,s}}{\sum_{b'\in\Omega}\sum_{s''\in S} v_{b',s'';a,s}}, \quad a \in \Sigma,\ s, s' \in S,\ b \in \Omega. \tag{37}$$

Proof. Let $S = \{s_1, \ldots, s_l\}$, $\Sigma = \{a_1, \ldots, a_{l_1}\}$, and $\Omega' = \{b_1, \ldots, b_{l_2}\}$. For each input-state pair (a_i, s_j), $1 \le i \le l_1$, $1 \le j \le l$, we have

$$\sum_{k=1}^{l_2}\sum_{m=1}^{l} \theta_{b_k,s_m;a_i s_j} = 1.$$

The parameters $\theta_{b_k,s_m;a_i s_j}$ with $1 \le k \le l_2$ and $1 \le m \le l$ appear in the log-likelihood function $\ell(\theta)$ as the partial sum

$$\ell_{i,j} = \sum_{k=1}^{l_2}\sum_{m=1}^{l} v_{b_k,a_m;a_i,s_j} \log(\theta_{b_k,a_m;a_i,s_j}).$$

Using $\theta_{b_{l_1},s_l;a_i,s_j} = 1 - \sum_{(b_k,s_m)\neq(b_{l_1},s_l)} \theta_{b_k,s_m;a_i,s_j}$, the partial derivative of $\ell_{i,j}$ with respect to $\theta_{b_k,s_m;a_i,s_j}$ becomes

$$\frac{\partial \ell_{i,j}}{\partial \theta_{b_k,s_m;a_i,s_j}} = \frac{v_{b_k,s_m;a_i,s_j}}{\theta_{b_k,s_m;a_i,s_j}} - \frac{v_{b_{l_2},s_l;a_i,s_j}}{1 - \sum_{(b_k,s_m)\neq(b_{l_2},s_l)} \theta_{b_k,s_m;a_i,s_j}}.$$

Equating this expression to 0 gives $\hat{\theta}_{b_k,s_m;a_i,s_j}$ as claimed. Thus the vector $\hat{\theta} = (\hat{\theta}_{b_k,s_m;a_i,s_j})$ is a critial point of the likelihood function.

Claim that this point maximizes the likelihood function; the proof idea goes back to Koski et al. [13]. Indeed, let $H(\theta) = -\sum_{i=1}^{n} \log \theta_i$ denote the entropy of a probability distribution $\theta = (\theta_1, \ldots, \theta_n)$ and let $D(\theta\|\theta') = \sum_{i=1}^{n} \theta_i \log\left(\frac{\theta_i}{\theta'_i}\right)$ denote the Kullback-Leibler measure between two probability distributions $\theta = (\theta_1, \ldots, \theta_n)$ and $\theta' = (\theta'_1, \ldots, \theta'_n)$. Then we have

$$\begin{aligned}
\ell(\theta) &= \sum_{a\in\Sigma}\sum_{s,s'\in S}\sum_{b\in\Omega} v_{b,s';a,s} \log \theta_{b,s';a,s} \\
&= \sum_{a\in\Sigma}\sum_{s,s',s''\in S}\sum_{b,b'\in\Omega} v_{b',s'';a,s}\hat{\theta}_{b,s';a,s} \log \theta_{b,s';a,s} \\
&= \sum_{a\in\Sigma}\sum_{s\in S} v_{a,s}\left(\sum_{b\in\Omega}\sum_{s'\in S} \hat{\theta}_{b,s';a,s} \log \theta_{b,s';a,s}\right)
\end{aligned}$$

$$
\begin{aligned}
&= \sum_{a\in\Sigma}\sum_{s\in S} v_{a,s}\left(\sum_{b\in\Omega}\sum_{s'\in S}\hat{\theta}_{b,s';a,s}\log\hat{\theta}_{b,s';a,s} - \hat{\theta}_{b,s';a,s}\log\frac{\hat{\theta}_{b,s';a,s}}{\theta_{b,s';a,s}}\right)\\
&= \sum_{a\in\Sigma}\sum_{s\in S} -v_{a,s}\left(H(\hat{\theta}_{a,s}) + D(\hat{\theta}_{a,s}\|\theta_{a,s})\right),
\end{aligned}
$$

where $v_{a,s} = \sum_{b'\in\Omega}\sum_{s''\in S} v_{b',s'';a,s}$, $\theta_{a,s} = (\theta_{b,s';a,s})$ and $\hat{\theta}_{a,s} = (\hat{\theta}_{b,s';a,s})$ for each input-state pair (a,s). Since the Kullback-Leibler measure is always non-negative [13], we obtain

$$
\begin{aligned}
\ell(\theta) &= \sum_{a\in\Sigma}\sum_{s\in S} -v_{a,s}\left(H(\hat{\theta}_{a,s}) + D(\hat{\theta}_{a,s}\|\theta_{a,s})\right)\\
&\leq \sum_{a\in\Sigma}\sum_{s\in S} -v_{a,s}H(\hat{\theta}_{a,s})\\
&= \sum_{a\in\Sigma}\sum_{s\in S} v_{a,s}\sum_{b\in\Omega}\sum_{s'\in S}\hat{\theta}_{b,s';a,s}\log\hat{\theta}_{b,s';a,s}\\
&= \sum_{a\in\Sigma}\sum_{s,s'\in S}\sum_{b\in\Omega} v_{b,s';a,s}\log\hat{\theta}_{b,s';a,s}\\
&= \ell(\hat{\theta}).
\end{aligned}
$$

This proves the claim and the result follows. □

A stochastic automaton is an abstract machine with an I/O interface. Therefore, suppose the sample data consist only of the input and output sequences, while the observer has no access to the state sequences. This problem can be tackled by the expectation-maximization (EM) algorithm. This is an iterative method to find the maximum posterior estimates of parameters in a statistical model with unobserved latent variables.

The aim is to estimate these probabilities by making use of a sample set. For this, let $A = (S,\Sigma,\Omega,p)$ be a stochastic automaton with the above setting and let $n \geq 1$. We assume that there is a collection $D = (d_1,\ldots,d_N)$ of N independent samples called *database*, where $d_r = (x_r, y_r) \in \Sigma^n \times \Omega^n$ denotes the r-th sample, $1 \leq r \leq N$. Then the joint probability of the sample d_r depending on the parameters is given by

$$
p_{X,Y|\Theta}(d_r \mid \theta) = \sum_{s\in S^{n+1}} p_{X,S,Y|\Theta}(x_r, s, y_r \mid \theta). \tag{38}
$$

The likelihood function $L = L_{X,Y}$ is given by

$$
L(\theta) = \prod_{r=1}^{N} p_{X,Y|\Theta}(d_r \mid \theta) = \prod_{(x,y)} p_{X,Y|\Theta}(x,y \mid \theta)^{u_{x,y}}, \tag{39}
$$

and the log-likelihood function $\ell = \ell_{X,Y}$ is

$$\ell(\theta) = \prod_{(x,y)} u_{x,y} \log p_{X,Y|\Theta}(x, y \mid \theta), \tag{40}$$

where $u_{x,y}$ is the number of times the sequence pair (x, y) is observed in the sample set. Therefore, we have

$$\sum_{(x,y)} u_{x,y} = N. \tag{41}$$

A version of the EM algorithm for stochastic automata is given by Alg. 2. Note that in the E-step, the marginal probabilities $p_{X,Y}(x, y|\theta)$ can be computed efficiently by the sum-product decomposition and in the M-step, the maximal estimate $\hat{\theta}$ can be calculated directly by Prop. 4.2.

Algorithm 2 EM algorithm for stochastic automata

Require: Stochastic automaton $A = (S, \Sigma, \Omega, p)$, joint probability function $p_{X,S,Y|\Theta}$, parameter space $\Theta \subseteq \mathbb{R}_{>0}^{l_1 l(l-1)(l_2-1)}$, integer $n \geq 1$, observed data $u = (u_{x,y}) \in \mathbb{N}^{l_1^n \times l_2^n}$

Ensure: Maximum likelihood estimate $\theta^* \in \Theta$

[Init] Threshold $\epsilon > 0$ and parameters $\theta \in \Theta$

[E-Step] Define matrix $U = (u_{x,s,y}) \in \mathbb{R}^{l_1^n \times l^{n+1} \times l_2^n}$ with

$$u_{x,s,y} = \frac{u_{x,y} \cdot p_{X,S,Y|\Theta}(x, s, y|\theta)}{p_{X,Y|\Theta}(x, y|\theta)}, \quad x \in \Sigma^n,\ s \in S^{n+1},\ y \in \Omega^n$$

[M-Step] Compute solution $\hat{\theta} \in \Theta$ of the likelihood function $\ell_{X,S,Y}$ with data set $U = (u_{x,s,y})$

[Compare] If $\ell_{X,Y}(\hat{\theta}) - \ell_{X,Y}(\theta) > \epsilon$, set $\theta \leftarrow \hat{\theta}$ and resume with E-step

[Output] $\theta^* \leftarrow \hat{\theta}$

The structure of stochastic automata allows an even more efficient implementation of the EM algorithm which amounts to a variant of the Baum-Welch algorithm. To see this, let $n \geq 1$ be an integer. Let $u = (u_{x,y}) \in \mathbb{N}^{l_1^n \times l_2^n}$ be a data vector, where $u_{x,y}$ is the number of times the sequence pair (x, y) is observed in the sample set. The full data vector $U = (u_{x,s,y}) \in \mathbb{N}^{l_1^n \times l^{n+1} \times l_2^n}$ is not available, where $u_{x,s,y}$ denotes the number of times the triple (x, s, y) is observed. The EM algorithm estimates in the E-step the counts of the full data

vector by the quantity

$$u_{x,s,y} = \frac{u_{x,y} \cdot p_{X,S,Y|\Theta}(x, s, y|\theta)}{p_{X,Y|\Theta}(x, y|\theta)}, \quad x \in \Sigma^n,\ s \in S^{n+1},\ y \in \Omega^n. \tag{42}$$

These counts provide the sufficient statistic v of the model and are used in the M-step to obtain updated parameter values based on the solution of the maximum likelihood problem. The expected values of the sufficient statistic v can be written in a way that leads to a more efficient implementation of the EM algorithm using dynamic programming.

For this, we introduce socalled forward and backward probabilities. The *forward probability*

$$f_{x,y,s}(i) = p_{X_1,\ldots,X_i,Y_1,\ldots,Y_i,S_{i+1}}(x_1, \ldots, x_i, y_1, \ldots, y_i, s), \tag{43}$$

where $s \in S$ and $1 \leq i \leq n$, is the joint probability that the prefixes $x_1 \ldots x_i$ and $y_1 \ldots y_i$ of the observed pair (x, y) having length i end in state s. We put $f_{x,y,s}(0) = 1/(l \cdot l_1)$. The *backward probability*

$$b_{x,y,s}(i) = p_{X_{i+1},\ldots,X_n,Y_{i+1},\ldots,Y_n|S_{i+1}}(x_{i+1}, \ldots, x_n, y_{i+1}, \ldots, y_n \mid s), \tag{44}$$

where $s \in S$ and $0 \leq i \leq n-1$, is the conditional probability that the suffixes $x_{i+1} \ldots x_n$ and $y_{i+1} \ldots y_n$ of the observed pair (x, y) having length $n - i$ start in state s.

Proposition 4.3. *In view of the sufficient statistic v, we have for all $s, s' \in S$, $a \in \Sigma$ and $b \in \Omega$,*

$$v_{b,s';a,s} = \sum_{y \in \Omega^n} \sum_{x \in \Sigma^n} \frac{u_{x,y}}{p(x, y|\theta)} \sum_{i=1}^{n} f_{x,y,s}(i-1) \cdot \theta_{b,s';a,s} \cdot b_{x,y,s'}(i), \tag{45}$$

where I_A denotes the indicator function of A; i.e., $I_A = 1$ if A is true and $I_A = 0$ otherwise.

Proof. For each state sequence $\sigma \in S^{n+1}$, we have

$$v_{b,s';a,s} = \sum_{y \in \Omega^n} \sum_{x \in \Sigma^n} \sum_{i=1}^{n} I_{(\sigma_i \sigma_{i+1} = ss')} \cdot u_{x,\sigma,y}.$$

Thus in view of (42), we obtain

$$v_{b,s';a,s} = \sum_{y \in \Omega^n} \sum_{x \in \Sigma^n} \frac{u_{x,y}}{p(x, y|\theta)} \sum_{i=1}^{n} \sum_{\sigma \in S^{n+1}} I_{(\sigma_i \sigma_{i+1} = ss')} \cdot p(x, \sigma, y|\theta).$$

The innermost term is the sum of all probabilities of triples (x, σ, y) for an input sequence x, an output sequence y and all state sequences σ such that $\sigma_i\sigma_{i+1} = ss'$. That is, observing the sequence pair (x, y) and a transition from state s to state s' at position i. Thus we have

$$\begin{aligned}
&\sum_{\sigma\in S^{n+1}} I_{(\sigma_i\sigma_{i+1}=ss')} \cdot p(x, \sigma, y|\theta) = \\
&\quad = \quad \mathbb{P}(X = x, Y = y, S_i = s, S_{i+1} = s') \\
&\quad = \quad \mathbb{P}(X_1 = x_1, \ldots, X_{i-1} = x_{i-1}, Y_1 = y_1, \ldots, Y_{i-1} = y_{i-1}, S_i = s) \\
&\qquad \cdot\mathbb{P}(Y_i = y_i, S_{i+1} = s' \mid X_i = x_i, S_i = s) \\
&\qquad \cdot\mathbb{P}(X_{i+1} = x_{i+1}, \ldots, X_n = x_n, Y_{i+1} = y_{i+1}, \ldots, Y_n = y_n \mid S_{i+1} = s') \\
&\quad = \quad f_{x,y,s}(i-1) \cdot \theta_{y_i,s';s,x_i} \cdot b_{x,y,s'}(i).
\end{aligned}$$

□

The probability $p(x, y|\theta)$ of the sequence pair (x, y) can be calculated based on the forward probabilities,

$$p(x, y|\theta) = \sum_{s\in S} f_{x,y,s}(n) \tag{46}$$

Note that the forward and backward probabilities can be recursively computed. For the sequence pair $(x, y) \in \Sigma^n \times \Omega^n$, consider the $l \times n$ matrices $F_{x,y} = (f_{x,y,s}(i))$ and $B_{x,y} = (b_{x,y,s}(i))$ of the forward and backward probabilities, respectively. The entries of the matrices $F_{x,y}$ and $B_{x,y}$ can be efficiently computed in an iterative manner,

$$f_{x,y,r}(0) = \frac{1}{l \cdot l_1}, \quad r \in S, \tag{47}$$

$$f_{x,y,r}(i) = \sum_{s\in S} f_{x,y,s}(i-1) \cdot \theta_{y_i,r;x_i,s}, \quad r \in S,\ 1 \le i \le n, \tag{48}$$

and

$$b_{x,y,r}(n) = 1, \quad r \in S, \tag{49}$$

$$b_{x,y,r}(i) = \sum_{s\in S} \theta_{y_{i+1},s;x_{i+1},r} \cdot b_{x,y,s}(i+1), \quad r \in S,\ 0 \le i \le n-1. \tag{50}$$

The calculation of the forward and backward probability matrices yields directly the sufficient statistic (Alg. 3). On the other hand, the EM algorithm needs to maintain the $l_1^n \times l^{n+1} \times l_2^n$ data set $U = (u_{x,s,y})$ from which the sufficient statistic can be obtained.

Algorithm 3 Baum-Welch algorithm for stochastic automata

Require: Stochastic automaton $A = (S, \Sigma, \Omega, p)$, joint probability function $p_{X,S,Y|\Theta}$, parameter space $\Theta \subseteq \mathbb{R}_{>0}^{l_1 l(l-1)(l_2-1)}$, integer $n \geq 1$, observed data $u = (u_{x,y}) \in \mathbb{N}^{l_1^n \times l_2^n}$

Ensure: Maximum likelihood estimate $\theta^* \in \Theta$

[Init] Threshold $\epsilon > 0$ and parameters $\theta \in \Theta$

[E-Step] Compute the sufficient statistic v as in Prop. 4.3

[M-Step] Compute solution $\hat{\theta} \in \Theta$ of the likelihood function $\ell_{X,S,Y}$ with data set $U = (u_{x,s,y})$

[Compare] If $\ell_{X,Y}(\hat{\theta}) - \ell_{X,Y}(\theta) > \epsilon$, set $\theta \leftarrow \hat{\theta}$ and resume with E-step

[Output] $\theta^* \leftarrow \hat{\theta}$

References

[1] P. D'Argenio, J.-P. Katoen, A theory of stochastic systems: Part I: stochastic automata, *Information and Control*, **203**, No. 1 (2005), 1-38. http://dx.doi.org/10.1016/j.ic.2005.07.001

[2] P. D'Argenio, J.-P. Katoen, A theory of stochastic systems: Part II: process algebra, *Information and Control*, **203**, No. 1 (2005), 39-74. http://dx.doi.org/10.1016/j.ic.2005.07.002

[3] D. Barber, *Bayes Reasoning and Machine Learning*, Cambridge Univ. Press, Cambridge (2012).

[4] D. Blackwell, L. Breiman, A.J. Thomasian, The capacitites of certain channel classes under random coding, *Annals of Mathematical Statistics*, **31**, No. 3 (1969), 558-567. http://dx.doi.org/10.1214/aoms/1177705783

[5] V. Claus, *Stochastische Automaten*, Teubner, Stuttgart (1971).

[6] J. von Neumann, Probabilistic logic and the synthesis of reliable organisms from unreliable components, in: Automata Studies, C. Shannon and J. McCarthy (eds), *Annals of Mathematical Studies*, **34**, Princeton Univ. Press, Princeton, NJ (1956). http://dx.doi.org/10.1515/9781400882618-003

[7] J. W. Carlyle, Reduced forms for stochastic sequential machines, *Journal Mathematical Analysis and Applications*, **7**, No. 2 (1963), 167-165. http://dx.doi.org/10.1016/0022-247X(63)90045-3

[8] L. Pachter, B. Sturmfels, *Algebraic Statistics for Computational Biology*, Cambridge Univ. Press, Cambridge (2005).

[9] L. R. Rabiner, A tutorial on hidden Markov models and selected applications, *Proceedings of the IEEE*, **77**, No. 2 (1989), 257-286. http://dx.doi.org/10.1109/5.18626

[10] M. O. Rabin, Probabilistic automata, *Information and Control*, **6**, No. 3 (1963), 230-245. http://dx.doi.org/10.1016/S0019-9958(63)90290-0

[11] M. O. Rabin, D. Scott, Finite automata and their decision problems, *IBM Journal Research Development*, **3**, No. 3 (1959), 114-125. http://dx.doi.org/10.1147/rd.32.0114

[12] G. Ricciardi, R. Pieraccini, E. Bocchieri, Stochastic automata for language modeling, *Computer Speech and Language*, **10** (1996), 265-293. http://dx.doi.org/10.1006/csla.1996.0014

[13] T. Koski, J. M. Noble, *Bayesian Networks*, Wiley, New York (2009).

[14] L. Rabiner, B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, Englewood Cliffs, NJ (1993).

[15] A. Salomaa, *Theory of Automata*, Pergamon Press, Oxford (1969).

[16] C. E. Shannon, The mathematical theory of communication, *Bell System Technical Journal*, **5**, No. 1 (1948), 379-423. http://dx.doi.org/10.1002/j.1538-7305.1948.tb01338.x

[17] P. H. Starke, Stochastische Ereignisse und Wortmengen, *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, **12** (1966), 61-68. http://dx.doi.org/10.1002/malq.19660120108

[18] M. Thathachar, Stochastic automata and learning systems, *Sadhana*, **15** (2009), 263-281. http://dx.doi.org/10.1007/BF02811325

[19] A. J. Viterbi, Error bounds for convolutional codes and an asymptotic optimum decoding algorithm, *IEEE Transactions on Information Theory*, **13**, No. 2 (1967) 260-269. http://dx.doi.org/10.1109/TIT.1967.1054010

[20] K.-H. Zimmermann, *Algebraic Statistics*, TubDok, Hamburg, Germany (2016).