Vesa-Matti Mäntylä

Discrete hidden Markov models with application to isolated user-dependent hand gesture recognition

TECHNICAL RESEARCH CENTRE OF FINLAND ESPOO 2001

VTT PUBLICATIONS 449

Discrete hidden Markov models with application to isolated user-dependent hand gesture recognition

Vesa-Matti Mäntylä
VTT Electronics

TECHNICAL RESEARCH CENTRE OF FINLAND
ESPOO 2001

ISBN 951–38–5875–8 (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–5876–6 (URL:http://www.inf.vtt.fi/pdf/)
ISSN 1455–0849 (URL:http://www.inf.vtt.fi/pdf/)

JULKAISIJA – UTGIVARE – PUBLISHER

Valtion teknillinen tutkimuskeskus (VTT), Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 4374

Statens tekniska forskningscentral (VTT), Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax (09) 456 4374

Technical Research Centre of Finland (VTT), Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Elektroniikka, Verkkoteknologiat, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 551 2111, faksi (08) 551 2320

VTT Elektronik, Nätteknologier, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 551 2111, fax (08) 551 2320

VTT Electronics, Networking Research, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland
phone internat. + 358 8 551 2111, fax + 358 8 551 2320

Otamedia Oy, Espoo 2001

Mäntylä, Vesa-Matti. Discrete hidden Markov models with application to isolated user-dependent hand gesture recognition. Espoo 2001. Technical Research Centre of Finland, VTT Publications 449. 104 p.

Keywords discrete hidden Markov models, hand gesture recognition, stochastic processes, discrete Markov chains, Bayes classification

Abstract

The development of computers and the theory of doubly stochastic processes, have led to a wide variety of applications of the hidden Markov models (HMMs). Due to their computational efficiency, discrete HMMs are often favoured. HMMs offer a flexible way of presenting events with temporal and dynamical variations. Both of these matters are present in hand gestures, which are of increasing interest in the research of human-computer interaction (HCI) technologies. The exploitation of human-to-human communication modalities has become actual in HCI applications. It is even expected, that the existing HCI techniques become a bottleneck in the effective utilization of the available information flow.

In this work it is given mathematically uniform presentation of the theory of discrete hidden Markov models. Especially, three basic problems, scoring, decoding and estimation, are considered. To solve these problems it is presented forward and backward algorithms, Viterbi algorithm, and Baum-Welch algorithms, respectively.

The second purpose of this work is to present an application of discrete HMMs to recognize a collection of hand gestures from measured acceleration signals. In pattern recognition terms, it is created an isolated user-dependent recognition system. In the light of recognition results, the effect of several matters to the optimality of the recognizer is analyzed.

Preface

The main ideas for this thesis have been induced by the work done in the Networking Research area of VTT Electronics, during last two years. This work has been submitted to the University of Oulu, Faculty of Natural Sciences, as part of the qualification for the degree of Licenciate of Philosophy. I want to express my gratitude to VTT Electronics for making possible the research and writing processes.

Special thanks to Tapio Peltola, who has given valuable help concerning the theory of hidden Markov models and especially their practical implementation. I would also like to thank Prof. Tapio Seppänen, from the University of Oulu, for the valuable discussions about pattern recognition matters concerning the development of the recognition system. During the research and writing processes Esa Tuulari, Pekka Ruuska and Tapio Frantti have given encouraging comments, also. For this, special thanks to them.

I would like to thank the supervisors of this thesis, PhD Erkki Laitinen from the Department of Mathematical Sciences of the University of Oulu and Prof. Petri Mähönen from VTT Electronics, for their valuable comments on writing this thesis.

This work has been funded partially by Tekes, Technical Development Centre Finland in the scope of the ITEA 99002 BEYOND project. In this connection, I want to express thanks to Mikko Kerttula.

Oulu, August 2000

Vesa-Matti Mäntylä

Contents

1 Introduction

1.1 Historical perspective

The roots of the theory of the *hidden Markov models* (HMMs) can be traced to the 1950s, when statisticians were studying the problem of characterizing random processes for which the incomplete observations were available. Their approach was to model the problem as a ”doubly stochastic process” in which the observed data were thought to be result of having passed the ”hidden” process through a ”censor” that produced the ”observed” process. The characterizing of the both processes was to be performed using only the observed process. This led to the discovery of the *expectation-maximization* (EM) algorithm, which is a general-purpose algorithm for maximum likelihood estimation in a wide variety of situations, known as incomplete-data problems. In the late 1960s and early 1970s, Baum and his colleagues worked with the special type of probabilistic functions of Markov chains, later known as hidden Markov models. As a result of this, the *forward-backward algorithm* or the *Baum-Welch reestimation algorithm* for the parameter estimation of the HMMs was revealed in a series of papers [4], [5], [6], [7], [8]. This algorithm can be seen as an early version of the EM algorithm and it still is the basis of reestimation algorithms used in the applications of the HMMs. [11][31]

The Baum-Welch algorithm offered an computationally effective solution for the model parameter estimation or the training problem of HMMs. This, together with the development of computers, has led to a wide variety of practical applications of the HMMs. In the last two decades *automatic speech recognition* (ASR) has been the major application area for HMMs. A great deal of this work is due to Rabiner and his colleagues. Their classical works [36], [37], and [38] have made greatly known the basic theory of HMMs and are referred also in other application fields. Recently, HMMs have been applied to a variety of applications outside of speech recognition, such as *hand gesture* [32] and *handwriting recognition* [12], pattern recognition in *molecular biology* [3], *fault-detection in dynamic systems* [43], and the *modeling of error burst characteristics of communication channels* [46].

1.2 Motivation of hand gesture recognition

With the massive influx of computers in everyday life, *human-computer interaction* (HCI), has become an increasingly important part of our daily lives. It is widely believed that as the computing, communication, and display technologies progress even further, the existing HCI techniques become a bottleneck in the effective utilization of the available information flow. In recent years, there has been an increasing interest in trying to introduce human-to-human communication modalities into HCI. This includes a class of techniques based on the hand gestures. Human hand gestures are an expressive means of non-verbal interaction among people ranging from simple pointing actions to the more complex ones expressing feelings and allowing the human communication. The exploitation of the hand gestures in HCI requires the means by which the gestures can be interpreted by computers. This process includes reasonable measuring and modeling of hand gestures. [34]

The HCI interpretation of gestures requires that dynamic and/or static configurations of the human hand, be measurable by the machine. The most conventional approaches to measure hand gestures have employed cybergloves [23]. In recent years, the computer vision community has also shown a lot of interest recently in the recognition of human actions and gestures. This is strongly enlightened by the review of Pavlovic et al. [34]. Sequential images of hand gestures are used in [18], [24], [30] and [32]. Alternative approaches for hand gesture measurement have also been presented: a mouse is used as a two-dimensional hand gesture input device in [49], whereas accelerometers have been employed in [15], [41], and [47].

In previously mentioned systems for hand gesture recognition, the applications are directly related to HCI or at least to HCI in computer operated systems:
- the presentation of words of sign language in [18],
- robot teleoperation and programming in [23] and [49],
- visually mediated control system to control an active teleconferencing camera in [30],
- graphic editor system operated by hand gestures in [32],
- arm gesture recognition system as an alternative method of computer input for people with severe speech and motor impairment in [15],

- and musical performance control and conducting recognition systems in [41] and [47].

HMMs offer a flexible way of presenting events with temporal and dynamical variations. These advantages have been employed in [23], [29], [32], [49], and [47]. A modified HMM, called partly-hidden Markov model is used in [18] to hand gesture recognition. Other kind of statistical methods are applied in [30] and [41].

1.3 Pattern recognition concepts

The goal of pattern recognition (PR) is to classify *objects* into a number of *classes* or categories. These objects may be images, signal waveforms or any type of measurements that are to be classified, which are referred as *patterns*. In recent decades there have happened rapid developments in computer technology and automation in industrial processes, increasing need for information handling and retrieval. These facts have changed PR from a theoretical research area of statistics into the high edge of modern engineering applications and research. PR techniques have become an important *component of intelligent systems* and are used for *data preprocessing and decision making*.

Pattern recognition does not include only one approach. It is a broad collection of often loosely related knowledge and techniques. This is clearly illustrated by the large overlap of PR with other areas, such as: *signal processing and systems, artificial intelligence, neural modeling optimization/ estimation theory, automata theory, fuzzy sets, structural modeling and formal languages*. In the historical course, the two main approaches to PR are *the statistical* (or decision theoretic) and *the syntactic* (or structural) *approaches*. But the recent advances in the use of *neural networks* (NNs) have created a third and remarkable approach to PR techniques. [42]

Nowadays, there exist a wide range of applications of PR techniques. As an example, for machine vision systems PR is of great importance. Typical applications of machine vision system are met in the manufacturing industry, for *automated visual inspection* or for *automation in the assembly line*. In inspection, manufactured objects are classified into ”defect” or

”nondefect” classes according to the on line PR system analysis made from images of objects captured by control cameras. In an assembly line, objects are located and classified in one of a number of classes, which are known in advance. *Character recognition* is also an classical area for PR applications in automation and information handling. For example, an *optical character recognition system* (OCR) consists of a light source, a scan lens, a document transport, and a detector. At the output of the light-sensitive detector, light intensity variation is translated into ”numbers” and an image array is formed. After this, the usage of image processing techniques leads to line and character segmentation. Finally, the PR software recognizes the characters, or in other words, classifies each character into correct ”letter”, ”number”, or ”punctuation” class. This gives major advances in further electronic processing and storing recognized ASCII characters is much more efficient than storing scanned document images. Other important areas for PR applications are: *image preprocessing, segmentation, and analysis, seismic, analysis, radar signal classification/ analysis,face and gesture recognition, speech recognition, fingerprint identification, handwriting analysis, electrocardiographic signal analysis and medical diagnosis*. [26][42][44]

In the definition of PR, we presented the concept of pattern, the object of classification. It is equally important to represent the concept of *feature*, which in broad sense, means any extractable measurement used. Signal intensities are examples of low-level features. Features may also be *symbolic*, *numerical*, or both. A symbolic feature could be color, whereas mass, measured in grams, is an example of a numeric feature. Features may also result from a process of feature extraction.In feature extraction it is very often decreased the dimension of the measurement vector with, for example, principal component analysis (PCA) [44, pp. 184 - 189] of the data. Feature extraction can also produce higher level entities computed from the measurement vectors. The key in this process, is to select and to extract features that are computationally feasible, lead good classification results and possibly reduce the raw measurements into a manageable amount of information without discarding essential information [42]. Feature extraction is not an obligatory part of the PR system. In many cases, a considerable amount of computation is dedicated to action, which is called *preprocessing*. By this it is often meant filtering or transforming of the raw data to aid computational feasibility and feature extraction and minimize noise. The typical structure of a PR system is presented in Figure 1.

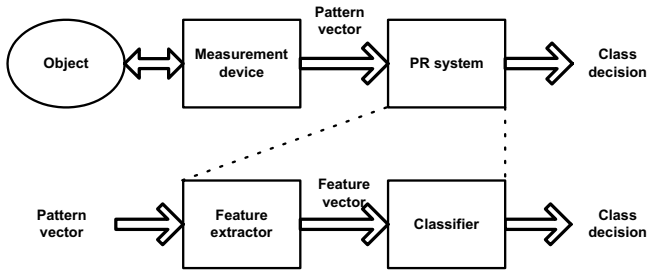

Figure 1. A typical structure of a PR system.

Usually, there are available data that have been classified or described in advance, which makes it possible to train the classifier or the descriptor. This is known as *supervised pattern recognition*. Another type of PR tasks for which training data, with known class labels, are not available. In this case, we are given a set of feature vectors and the goal is to unravel the similarities, and cluster, in some sense, similar vectors together. This is called the *unsupervised PR* or *clustering*. Such tasks appear in many applications in social sciences and engineering, such as *remote sensing, image segmentation, and image and speech coding*.

1.4 Scopes of the research

Previous sections have shown that, there has been a wide interest in applying the HMMs in different kind of pattern recognition problems. This together with the demand of computational tractability have given a good reason to set the focus of the methodological research on discrete HMMs. There exist several practical works like [28] or [37], which give a good insight in the theory of discrete HMMs from more engineering like perspective. The work of Timo Koski [20] gives quite wide but not so accurate analysis of the theory of discrete HMMs in the context of computational biology. At the same time he leaves open questions to work out by the reader. The unavailability of the uniform presentation of the theory of discrete HMMs and related algorithms with accuracy of satisfactory gives one of the main scopes of this research:
- to collect a mathematically uniform and satisfactory presentation of the theory of discrete hidden Markov models growing from the background of

discrete Markov chains.
Discrete Markov chains are considered in Chapter 2, while discrete hidden Markov models with related algorithms are presented in Chapter 3. The principle of Bayes classification is considered in Chapter 4.

The importance of hand gesture recognition was enlightened previously. In Chapter 5, it is presented an application of discrete HMMs to recognize a collection of hand gestures from measured acceleration signals. All signals corresponding to an individual gesture are segmented in advance, and the measurements are collected from one person. Thus, in pattern recognition terms, the created recognition system is called an *isolated user-dependent recognition system*. The left-right type of HMM has the desirable ability of modeling signals whose properties change over time in a successive manner, like speech. The same fact is present in hand gestures, too. This is practically verified in [23], [47], [32], and [49]. It is clear by these works, that the topology of HMMs leaves no questions. In speech recognition, it has been developed standard ways of parametrizing the acoustic signals, LPC or cepstral analysis. This is not the case, in hand gesture recognition, especially with acceleration signals. Several succeeding parametrizing methods have been proposed in, for example, [23], [47], [32], and [49] for hand gesture recognition with HMMs. In [47] this has been performed to the accelerometer data. The parametrization or feature extraction of the measured signals lays the basis for the design of the HMM recognizer. Different feature space leads to different choices with vector quantization:
- *codebook generation, and*
- *reasonable size of the codebook,*
or with model parameters:
- *number of states per model, and*
- *sufficient amount of training sequences.*
All this is performed to create a reliable recognizer, in other words, a recognizer with recognition results high enough. In the light of the recognition results, that are presented in Chapter 6, previously presented problems are considered in Chapter 7. [37]

2 Stochastic processes and discrete Markov chains

2.1 Probabilistic prerequisites

One of the most important application of the probability, is as a measure of randomness. By a classical *random experiment*, it is meant an experiment, whose outcome is not known in advance but for which the set of all possible individual outcomes is known. This set is called the *sample space* Ω. Individual outcomes are called *sample points* or *elementary events*. An event, denoted by A, is a subset of a sample space Ω. An event A is said to occur, if the random experiment is performed and the observed outcome $x \in A$.

In the following, it is presented the essential concepts and theorems of probability calculus for this work. For the proofs of the theorems see [45]. [14][40]

Definition 1 *A δ-algebra F on set Ω, is a set of all subsets on Ω satisfying conditions*
(1) $\Omega \in F$;
(2) $A \in F \Longrightarrow A^c \in F, where\ A^c = \Omega - A$;
(3) $A_1,\ A_2,\ ... \in F \Longrightarrow \bigcup_n^\infty A_n \in F$.

From conditions (2) and (3), it follows immediately that the empty set $\phi = \Omega^c \in A$ and thus the intersection $\bigcap_n^\infty A_n = \left(\bigcup_n^\infty A_n^c\right)^c \in F$.

Definition 2 *A probability measure P on the pair (Ω, F) is a function $P : F \longrightarrow [0, \infty[$, satisfying following properties:*
(1) $P(A) \geq 0$, *for all* $A \in F$;
(2) $A \cap B = \phi \Longrightarrow P(A \cup B) = P(A) + P(B)$;
(3) $P(\Omega) = 1$;
(4) $B_1 \supset B_2 \supset ... \supset B_n \supset ...$ *and* $\bigcap_n^\infty B_n = \phi \Longrightarrow \lim_{n \longrightarrow \infty} P(B_n) = 0$.

The triple (Ω, F, P) is called a *probability space* or a *probability field*.

Theorem 1 *Let triple (Ω, F, P) be a probability space. Then*
(1) $P(\phi) = 0$;
(2) $P(A) + P(A^c) = 1$;
(3) $P(A \cup B) = P(A) + P(B) - P(A \cap B)$*, for all* $A, B \in F$;
(4) $A \subset B \Longrightarrow P(A) \leq P(B)$*, for all* $A, B \in F$.

Definition 3 *Events $A_1,\ A_2,\ \ldots,\ A_n$ on a probability space (Ω, F, P) are called independent, if the joint probability equals*

$$P(A_{i_1} \cap A_{i_2} \cap \ldots \cap A_{i_j}) = P(A_{i_1}) \cdot P(A_{i_2}) \cdot \ldots \cdot P(A_{i_j}), \qquad (1)$$

for all $1 \leq i_1 < i_2 < \ldots < i_j \leq n$ *and* $j = 2,\ 3,\ \ldots,\ n$.

Definition 4 *Let A and B be events on a probability space (Ω, F, P). The conditional probability of A, given B, is defined by*

$$P(A|\, B) = \frac{P(A \cap B)}{P(B)}, \qquad (2)$$

if $P(B) > 0$.

Theorem 2 *Let A, B and C be events on a probability space (Ω, F, P). If $P(B) > 0$, then* $P(A \cap B|\, C) = P(A|\, B \cap C)P(B|\, C)$.

Proof. By previous definition the conditional probability becomes

$$\begin{aligned} P(A \cap B|\, C) &= \frac{P(A \cap B \cap C)}{P(C)} \\ &= \frac{P(A \cap (B \cap C))}{P(C)} \\ &= \frac{P(A|\, B \cap C)P(B \cap C)}{P(C)} \\ &= \frac{P(A|\, B \cap C)P(B|\, C)P(C)}{P(C)} \\ &= P(A|\, B \cap C)P(B|\, C). \end{aligned}$$

Theorem 3 *Let $B_1,\ B_2,\ \ldots,\ B_n$ be events on a probability space (Ω, F, P), with $B_i \cap B_j = \phi$, for all $i \neq j$, and $\bigcup_{i=1}^{n} B_i = \Omega$. If the event $A \in F$, then*

the conditional probability equals

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{\sum\limits_{k=1}^{n} P(B_k)P(A|B_k)}, \tag{3}$$

for all $i \in \{1,\ 2,\ \ldots\ ,\ n\}$.

Probabilities $P(B_i)$ are called *a priori* probabilities and probabilities $P(B_i|A)$ are called *a posteriori probabilities*, for all $i = 1,\ 2,\ \ldots\ , n$.

Definition 5 *A random variable* X *on a probability space* (Ω, F, P) *is a function* $X : \Omega \longrightarrow R$, *satisfying* $\{\omega \in \Omega |\ X(\omega) \leq x\} \in F$, *for all* $x \in R$.

Theorem 4 *Let* $X_i,\ i = 1, 2, \ldots , n$, *be discrete random variables on a probability space* (Ω, F, P). *Then the linear combination*

$$c_1X_1 + c_2X_2 + \ldots + c_nX_n$$

is a random variable on (Ω, F, P), *for all* $c_i \in R,\ i = 1,\ 2,\ \ldots\ ,\ n$.

Definition 6 *A random variable* $X : \Omega \longrightarrow R$ *on a probability space* (Ω, F, P), *is called discrete, if the set* $D = \{x|\ P(X = x) > 0\}$ *consists of either a finite set, say* $x_1,\ x_2,\ \ldots,\ x_J$, *or an infinite countable set, say* $x_1,\ x_2,\ \ldots$, *and in addition the sum equals*

$$\sum_{x_i \in D} P(X = x_i) = 1. \tag{4}$$

Definition 7 *A distribution* f_X *of a discrete random variable* X *on probability space* (Ω, F, P), *is defined by*

$$f_X(x_i) = P(X = x_i), \tag{5}$$

for all $x_i \in D = \{x|\ P(X = x) > 0\}$.

Theorem 5 *If* f_X *is a distribution of a discrete random variable* $X : \Omega \longrightarrow R$, *then*
(1) $f_X(x) \geq 0$, *for all* $x \in R$ *and*
$f_X(x) > 0 \Leftrightarrow x \in \{x_1,\ x_2,\ \ldots\ ,\ x_n,\ \ldots\} \subset R$;
(2) $\sum\limits_{x_i \in D} f_X(x_i) = 1$.

Definition 8 *Let $X : \Omega \longrightarrow R$ be a discrete random variable on a probability space (Ω, F, P). The expected value or mean of X, $\mu = E(X)$, is defined by*

$$\mu = E(X) = \sum_{x_i \in D} x_i P(X = x_i), \tag{6}$$

provided that the sum equals

$$\sum_{x_i \in D} |x_i| P(X = x_i) < \infty, \tag{7}$$

where $D = \{x |\ P(X = x) > 0\}$.

Condition 7 is called the *absolute convergence.*

Theorem 6 (Linearity of the expected value) *Let $X_i,\ i = 1,\ 2,\ \ldots,\ n$, be discrete random variables on a probability space (Ω, F, P), for which the expected values equal $\mu_i = E(X_i),\ i = 1,\ 2,\ \ldots,\ n$. Then the expected value of the random variable $c_1X_1 + c_2X_2 + \ldots + c_nX_n$ equals*

$$E\left(c_1X_1 + c_2X_2 + \ldots + c_nX_n\right) = c_1E\left(X_1\right) + c_2E(X_2) + \ldots + c_nE(X_n)\,,$$

for all $c_i \in R,\ i = 1,\ 2,\ \ldots,\ n$.

Definition 9 *Let $X : \Omega \longrightarrow R$ be a discrete random variable on a probability space (Ω, F, P). The conditional expectation of X, on the condition $A \in F,\ P(A) > 0$, is defined by*

$$E(X |\, A) = \sum_{x_i \in D} x_i P(X = x_i |\, A), \tag{8}$$

provided that the sum equals

$$\sum_{x_i \in D} |x_i| P(X = x_i |\, A) < \infty, \tag{9}$$

where $D = \{x |\ P(X = x) > 0\}$.

By the linearity of expectation and previous definition, the linearity of the conditional expectation is clear.

Definition 10 *Let $X : \Omega \longrightarrow R$ be a discrete random variable on a probability space (Ω, F, P). The variance of X, $\delta^2 = Var(X)$, is defined by*

$$\delta^2 = Var(X) = E[(X - \mu)^2] = \sum_{x_i \in D} (x_i - \mu)\, P(X = x_i), \tag{10}$$

where μ is the expected value of X and $D = \{x |\ P(X = x) > 0\}$.

2.2 Markov chains

Markov process is named after A. A. Markov who introduced the concept in 1907 with a discrete time and finite number of states. The denumerable case was launched by Kolmogorov in 1936, followed closely by Doeblin whose contributions pervade all parts of the Markov theory. Markov chains have been used a good deal in applied probability and statistics. In these applications one is generally looking for something considerably more specific or rather more general. In the former category belong for example finite chains and birth-and-death processes, whereas in the latter belong various models involving a continuous state space subject to some discretization such as queuing problems. [10]

Definition 11 *A stochastic process with state space S is a sequence $\{X_t\}_{t\in T}$ of random variables $X_t \in S$ defined on the same probability space (Ω, F, P).*

The set T is called the *parameter set*. It is customary to think of the index $t \in T$ as representing time. Thus state $X_t,\ t \in T$, is thought as the state of the process at time t. If T is finite or countable, the process is said to be a *discrete-time* or *discrete parameter process*.

Definition 12 *A stochastic process $\{X_t\}_{t\in T}$ with state space S on a probability space (Ω, F, P) is a Markov process, if for any set of $n+1$ values $t_1 < t_2 < ... < t_n < t_{n+1},\ t_i \in T,\ i = 1,\ 2,\ ...\ ,\ n+1$, and any set of states $\{x_1,\ x_1,\ ...\ ,\ x_{n+1}\} \subset S$, the conditional probability equals*

$$\begin{aligned} & P\left(X(t_{n+1}) = x_{n+1} |\ X(t_1) = x_1,\ X(t_2) = x_2,\ ...\ ,\ X(t_n) = x_n\right) \\ = \ & P\left(X(t_{n+1}) = x_{n+1} |\ X(t_n) = x_n\right). \end{aligned} \tag{11}$$

Condition 11 makes it clear that the future of the process depends only on the present state and not upon the history of the process. In other words, the history of the process is summarized in the present state.

Definition 13 *A Markov process $\{X_t\}_{t\in T}$ with state space S on a probability space (Ω, F, P) is called a Markov chain, if the state space S is finite or countable, in other words discrete.*

With respect to time and state space discontinuity or continuity, Markov processes are classified to four types: discrete-time Markov chain, continuous-time Markov chain, discrete-time Markov process, and continuous-time Markov process.

For now on, we are dealing with discrete-time Markov chains with finite state space S. Thus, for simplicity, it is set the parameter set (or time) $T = N$, and the state space $S = \{1,\ 2,\ \ldots\ , J\}$, where $J \in Z_+,\ J < \infty$. By these assumptions, the discrete-time Markov chain is denoted by $\{X_n\}_{n=0}^{\infty}$. Unless otherwise stated, all random variables are assumed to be discrete and defined on the probability space (Ω, F, P).

For a discrete-time Markov chain it is useful to think of the process as making *state transitions* at times $n \geq 1$. The conditional Markov probabilities

$$p_{ij} = P(X_{n+1} = j |\ X_n = i),\ n \geq 0,\ i,\ j \in S \tag{12}$$

are assumed to be independent of the time-instant n and they are called *stationary one-step transition probabilities*. In the case, that this probability is not defined, it is set $p_{ij} = 0$. Property 12 is also known as *homogenousity property* and Markov chain satisfying this property is called *a homogenous Markov chain*.

The probabilities p_{ij} are usually presented in matrix form

$$P = (p_{ij})_{i=1,j=1}^{J,J} \tag{13}$$

or

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1J} \\ p_{21} & p_{22} & \cdots & p_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ p_{J1} & p_{J2} & \cdots & p_{JJ} \end{pmatrix}. \tag{14}$$

Matrix P is to be called a *transition matrix*. Row i of P is the conditional probability distribution of X_n given that state $X_{n-1} = i$. It is natural to expect the following stochastic constraints to the matrix P:

$$p_{ij} \geq 0, \tag{15}$$

$$\sum_{j=1}^{J} p_{ij} = 1, \tag{16}$$

for all $i,\ j = 1,\ \ldots,\ J$.

The discrete Markov chain $\{X_n\}_{n=0}^{\infty}$ starts in an initial state $X_0 \in S$, and makes a state transition at the next time step in the sequence. The probabilities

$$\pi_i(0) = P(X_0 = i),\ i \in S, \tag{17}$$

are called the *initial state probabilities* and define the *initial state distribution* $\pi(0) = (\pi_1(0),\ \pi_2(0),\ \ldots,\ \pi_J(0))$ of the Markov chain. It is natural a demand, that the sum equals

$$\sum_{i=1}^{J} \pi_i(0) = 1. \tag{18}$$

By the definitions of transition matrix P and initial distribution $\pi(0)$, it is clear that for each Markov chain with stationary transition probabilities, matrix P and vector $\pi(0)$ are unequivocally defined.

Theorem 7 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain with initial distribution $\pi(0) = (\pi_1(0),\ \pi_2(0),\ \ldots,\ \pi_J(0))$ and transition matrix $P = (p_{ij})_{i=1,j=1}^{J,J}$. Then, the joint probability equals*

$$\begin{aligned} & P\left(X_0 = j_0, X_1 = j_1, ..., X_{k-1} = j_{k-1}, X_k = j_k\right) \\ = \ & P(X_0 = j_0) \prod_{i=1}^{k} P(X_{j_i} | X_{j_{i-1}}) \\ = \ & \pi_{j_0}(0) \prod_{i=1}^{k} p_{j_{i-1} j_i}. \end{aligned} \tag{19}$$

Proof. See [10, pp. 5 - 6].

Example 1 (Binary information source) A Markov information source is a sequential mechanism for which the chance that a certain symbol is produced depends upon the preceding symbol. If the symbols may get two distinct values, for example 0 and 1, we talk about a binary Markov source. If at some stage symbol 0 is produced, then at the next stage symbol 1 will be produced with probability p and symbol 0 will be produced with probability $1-p$. If a symbol 1 is produced, then at the next stage symbol 0 will be produced with probability q and 1 will be produced with probability

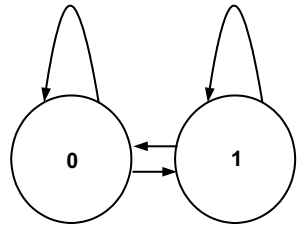

Figure 2. A state diagram of a binary information source.

$1-q$. Assuming a two-state Markov chain, we have one state corresponding the production of symbol 0 and another state corresponding to symbol 1. For this Markov chain we get a transition matrix

$$\begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}. \tag{20}$$

A Markov chain is usually illustrated by a state diagram showing the possible transitions between states. See Figure 2.

Example 2 (Repeaters) Binary digits 0 and 1 are transmitted over a sequential system of n repeaters with noise. Let the probability that a digit entering the k:th repeater is remained unchanged, be p, and let the probability for the reverse case be $1-p$. These probabilities are assumed to be the same for all repeaters. It is also assumed that all repeaters work independently of each other during the transmission of digits. Let X_0 denote a digit that enters the first repeater and X_k, $k = 1, \ldots, n$ be the digit that leaves the k:th repeater. It can be shown that the sequence $X_0, X_1, \ldots, X_n$ satisfies the Markov property.

Proof. By the definition of conditional probability

$$\begin{aligned} &P\left(X_k = j_k \middle| X_0 = j_0, X_1 = j_1, \ldots, X_{k-1} = j_{k-1}\right) \\ &= \frac{P\left(X_0 = j_0, X_1 = j_1, \ldots, X_{k-1} = j_{k-1}, X_k = j_k\right)}{P\left(X_0 = j_0, X_1 = j_1, \ldots, X_{k-1} = j_{k-1}\right)} \end{aligned} \tag{21}$$

Since $X_0 = j_0$, the sequence of states $j_1, \ldots, j_{k-1}$, can be recovered from the sequence of differences $j_1 - j_0, \ldots, j_{k-1} - j_{k-2}$. Thus

$$P\left(X_0 = j_0, X_1 = j_1, \ldots, X_{k-1} = j_{k-1}\right)$$

$$= P\left(X_0 = j_0,\, X_1 - X_0 = j_1 - j_0,\, \ldots,\, X_{k-1} - X_{k-2} = j_{k-1} - j_{k-2}\right).$$

By the assumption of the independence of the repeaters the difference $X_k - X_{k-1}$ is independent of the difference $X_h - X_{h-1},\ h \neq k$. Hence

$$P\left(X_0 = j_0,\, X_1 - X_0 = j_1 - j_0,\, \ldots,\, X_{k-1} - X_{k-2} = j_{k-1} - j_{k-2}\right)$$

$$= P(X_0 = j_0) \prod_{i=1}^{k-1} P(X_i - X_{i-1} = j_i - j_{i-1})$$

and similarly

$$P\left(X_0 = j_0,\, X_1 = j_1,\, \ldots,\, X_{k-1} = j_{k-1},\, X_k = j_k\right)$$

$$= P(X_0 = j_0) \prod_{i=1}^{k} P(X_i - X_{i-1} = j_i - j_{i-1}).$$

By equation 21

$$P\left(X_k = j_k \middle|\, X_0 = j_0,\, X_1 = j_1,\, \ldots,\, X_{k-1} = j_{k-1}\right) = P(X_k - X_{k-1} = j_k - j_{k-1}).$$

However,

$$P(X_k - X_{k-1} = j_k - j_{k-1}) = \begin{cases} p & , \ if \ \ j_k - j_{k-1} = 0 \\ 1-p & , \ if \ \ j_k - j_{k-1} = 1 \ \ or \ \ j_k - j_{k-1} = -1. \end{cases}$$

Also,

$$P(X_k = j_k | X_{k-1} = j_{k-1}) = \begin{cases} p & , \ if \ \ j_k - j_{k-1} = 0 \\ 1-p & , \ if \ \ j_k - j_{k-1} = 1 \ \ or \ \ j_k - j_{k-1} = -1. \end{cases}$$

Thus,

$$P\left(X_k = j_k \middle|\, X_0 = j_0,\, X_1 = j_1,\, \ldots,\, X_{k-1} = j_{k-1}\right) = P\left(X_k = j_k \middle|\, X_{k-1} = j_{k-1}\right).$$

Theorem 8 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain. Then, conditional on, $X_m = i,\ m \geq 0$, the sequence $\{X_{m+n}\}_{n=0}^{\infty}$, is a Markov chain with initial state distribution $\delta_i = \{\delta_{ij} | j \in S\}$ and transition matrix P, where $\delta_{ij},\ i,\ j = 1,\ 2,\ \ldots,\ J$, is the Kronecker delta. In addition, the sequence $\{X_{m+n}\}_{n=0}^{\infty}$ is independent of the states $X_0,\ X_1,\ \ldots,\ X_m$.*

Proof. See [33, pp. 3 - 4].

Previous theorem shows, that given a Markov chain, a new chain with the same transition matrix is generated in every individual discrete time instant.

Following the definition of initial distribution $\pi(0)$, it is defined the state distribution $\pi(n)$, for state $X_n,\ n \geq 1$, as follows,

$$\pi(n) = (P(X_n = 1),\ P(X_n = 2),\ \ldots,\ P(X_n = J)),\ n \geq 1. \tag{22}$$

Theorem 9 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain. Then, for all integers $n \geq 0$, the state distribution equals*

$$\pi(n) = \pi(0)P^n. \tag{23}$$

Proof. See [33, pp. 3 - 4].

Corollary 10 *For the distribution of state X_n, it holds*

$$\pi(n) = \pi(n-1)P,\ n \geq 1. \tag{24}$$

Proof. Equation 23 implies, that

$$\pi(1) = \pi(0)P^1 = \pi(0)P. \tag{25}$$

From assumption

$$\pi(k) = \pi(k-1)P,\ k > 1, \tag{26}$$

and equation 23, it follows, that

$$\begin{aligned} \pi(k+1) &= \pi(0)P^{k+1} \\ &= \pi(0)P^k P \\ &= \pi(k)P. \end{aligned} \tag{27}$$

Thus, the state distribution equals

$$\pi(n) = \pi(n-1)P,\ n \geq 1. \tag{28}$$

Theorem 11 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain. Then for all integers $m,\ n \geq 0$, the conditional probability equals*

$$P(X_{m+n} = j|\ X_m = i) = P(X_n = j|\ X_0 = i) = p_{ij}(n),\ i,\ j \in S. \quad (29)$$

Proof. See [33, pp. 3 - 4].

The conditional probabilities

$$p_{ij}(n) = P(X_{m+n} = j|\ X_m = i),\ m,\ n \geq 0, \quad (30)$$

are called the *n-step transition probabilities* from state i to state j. These probabilities are usually presented in a matrix form, as follows

$$P(n) = (p_{ij}(n))_{i=1,j=1}^{J,J},\ n \geq 0. \quad (31)$$

For integer $n = 0$, it is set

$$p_{ij}(0) = \begin{cases} 1,\ if\ i = j \\ 0,\ if\ i \neq j \end{cases} = \delta_{ij},\ i,\ j\ =\ 1,\ 2,\ ...\ ,\ J. \quad (32)$$

Corresponding matrix becomes

$$P(0) = (\delta_{ij})_{i=1,j=1}^{J,J} = \begin{pmatrix} 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 1 \end{pmatrix} = I, \quad (33)$$

which is the identity matrix.

Theorem 12 (Chapman-Kolmogorov equations) *If $\{X_n\}_{n=0}^{\infty}$ is a Markov chain, then*

$$p_{ij}(n+1) = \sum_{k=1}^{N} p_{ik}(n)p_{kj}(1); \quad (34)$$

$$p_{ij}(m+n) = \sum_{k=1}^{N} p_{ik}(m)p_{kj}(n), \quad (35)$$

for all $m,\ n \geq 0$, and $i,\ j \in S$.

Proof. See [10, pp. 8 - 9].

Using matrix formalism, the equations 34 and 35 can be presented as follows

$$P(n+1) = P(n)P; \tag{36}$$
$$P(m+n) = P(m)P(n), \tag{37}$$

for all $m,\ n \geq 0$.

Theorem 13 *The n-step transition matrix equals*

$$P(n) = P^n, \tag{38}$$

for all $n \geq 0$.

Proof. Definition 33 implies, that

$$P(0) = I = P^0 \tag{39}$$

and from equation 36 it follows

$$P(1) = P(0+1) = P(0)P = IP = P^1. \tag{40}$$

Assumption

$$P(k) = P^k,\ k > 1, \tag{41}$$

and equation 36 imply, that

$$P(k+1) = P(k)P = P^k P = P^{k+1}. \tag{42}$$

Thus for all $n \geq 0$, the n-step transition matrix equals

$$P(n) = P^n. \tag{43}$$

Corollary 14 *The Chapman-Kolmogorov equation can be written as*

$$P(n+1) = P^{n+1} = P^n P; \tag{44}$$
$$P(m+n) = P^{m+n} = P^m P^n. \tag{45}$$

Example 3 We consider the situation in Example 2, where the transition matrix equals

$$P = \begin{pmatrix} 0.65 & 0.35 \\ 0.35 & 0.65 \end{pmatrix}.$$

It is to find the probability that a zero bit that is entered at the first stage is received as a zero bit by the fifth stage. The problem can be understood in the terms of n-step probabilities. It is actually asked the five-step probability $p_{00}(5)$. According to Theorem 13, the five-step probability matrix equals

$$\begin{aligned} P(5) &= P^5 \\ &= \begin{pmatrix} 0.5012 & 0.4988 \\ 0.4988 & 0.5012 \end{pmatrix}. \end{aligned}$$

Thus, the probability, that a zero bit will be transmitted through five stages as a zero is

$$p_{00}(5) = 0.5012.$$

[1, p. 223]

Definition 14 *A Markov chain $\{X_n\}_{n=0}^{\infty}$ is said to be stationary, if the probability*

$$P(X_n = j) = \pi_j, \tag{46}$$

for all $n \geq 0$ and $j \in S$.

In other words, for a stationary Markov chain, the probability of being in a state, is independent on the discrete time instant. Probabilities π_j, $j \in S$, define an *invariant distribution for transition matrix P* as follows,

$$\pi = (\pi_1,\ \pi_2,\ \ldots\ ,\ \pi_J), \tag{47}$$

if condition $P(X_0 = j) = \pi_j$ implies that $P(X_1 = j) = \pi_j$.

Theorem 15 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain with transition matrix P. The distribution $\pi = (\pi_1,\ \pi_2,\ \ldots\ ,\ \pi_J)$ is an invariant distribution if and only if*

$$\pi P = P, \tag{48}$$

with the restrictions

$$\sum_{j=1}^{J} \pi_j = 1; \tag{49}$$

$$\pi_j \geq 0. \tag{50}$$

Proof. Assume, that π is an invariant distribution. Then conditions 49 and 50 are clearly satisfied. Because π is an invariant distribution, there holds equation $\pi(0) = \pi(1) = \pi$. Thus equation 28 implies $\pi = \pi P$.

Assume now, that the distribution π satisfies the conditions 49 and 50. Let the initial distribution be denoted $\pi(0) = \pi$. Thus, the equation 28 implies

$$\pi(1) = \pi(0)P = \pi P = \pi, \tag{51}$$

which means that π is an invariant distribution.

Example 4 Suppose a telecommunication system, with three possible messages $1,\ 2,$ and 3, is a Markov chain with transition matrix

$$P = \begin{pmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.5 & 0.3 \\ 0.1 & 0.2 & 0.7 \end{pmatrix},$$

and initial distribution $\pi(0) = (0.5,\ 0.2,\ 0.3)$. According to Theorem 15, to find the invariant distribution $\pi = (\pi_1,\ \pi_2,\ \pi_3)$ it is to solve the following system of equations:

$$\begin{cases} \pi = \pi P \\ \pi_1 + \pi_2 + \pi_3 = 1 \end{cases}$$

$$\begin{cases} (P^T - I)\pi^T = 0 \\ \pi_1 + \pi_2 + \pi_3 = 1 \end{cases}$$

$$\begin{cases} \left\{ \begin{pmatrix} 0.6 & 0.2 & 0.1 \\ 0.2 & 0.5 & 0.2 \\ 0.2 & 0.3 & 0.7 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right\} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\ \pi_1 + \pi_2 + \pi_3 = 1 \end{cases}$$

$$\begin{cases} \begin{pmatrix} -0.4 & 0.2 & 0.1 \\ 0.2 & -0.5 & 0.2 \\ 0.2 & 0.3 & -0.3 \end{pmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\ \pi_1 + \pi_2 + \pi_3 = 1 \end{cases}$$

$$\begin{cases} -0.4\pi_1 + 0.2\pi_2 + 0.1\pi_3 = 0 \\ 0.2\pi_1 - 0.5\pi_2 + 0.2\pi_3 = 0 \\ 0.2\pi_1 + 0.3\pi_2 - 0.3\pi_3 = 0 \\ \pi_1 + \pi_2 + \pi_3 = 1 \end{cases},$$

for which the solution equals $\pi = (0.285714,\ 0.457143,\ 0.257143)$.

Theorem 16 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain with initial distribution $\pi(0)$ and transition matrix P. Suppose that π is the invariant distribution, then $\{X_{m+n}\}_{n=0}^{\infty}$ is also a Markov chain with initial distribution $\pi(0)$ and transition matrix P.*

Proof. See [33, p. 33].

Theorem 17 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain with finite state space S. Suppose for some state $i \in S$, that n-step probability*

$$p_{ij}(n) \to \pi_j,\ n \to \infty, \tag{52}$$

for all states $j \in S$. Then $\pi = (\pi_1,\ \pi_2,\ \ldots\ ,\ \pi_J)$ is an invariant distribution.

Proof. See [33, p. 33].

The result of the previous theorem serves to indicate a relationship between invariant distributions and n-step transition probabilities.

Definition 15 (Ergodicity) *A Markov chain $\{X_n\}_{n=0}^{\infty}$ is said to be ergodic, if there exists a probability distribution $e = (e_1,\ e_2,\ \ldots\ ,\ e_J)$ on the state space S such that the state distribution*

$$\pi(n) \to e,\ n \to \infty, \tag{53}$$

for any initial distribution $\pi(0)$.

According to the definition of a limit, the distribution e must be unique.

Theorem 18 *Let $\{X_n\}_{n=0}^{\infty}$ be an Markov chain with initial distribution $\pi(0)$ and transition matrix P. If*

$$\pi(n) \to e,\ n \to \infty, \tag{54}$$

where $e = (e_1,\ e_2,\ \ldots\ ,\ e_J)$ is a probability distribution. Then e is an invariant distribution.

Proof. Equations 28 and 54 imply

$$\begin{aligned} e &= \lim_{n\to\infty} \pi(n) = \lim_{n\to\infty} \pi(n+1) = \\ &= \lim_{n\to\infty} (\pi(n)P) = \left(\lim_{n\to\infty} \pi(n)\right) P = eP. \end{aligned}$$

The interchange of order taking matrix multiplication and limit is permissible, since the state space S is assumed to be finite.

Previous theorem shows the identity between the limiting distribution and the invariant distribution for an ergodic Markov chain. On the other hand, it states that an ergodic Markov chain becomes asymptotically stationary.

Theorem 19 *Let $\{X_n\}_{n=0}^{\infty}$ be an Markov chain. If for all pairs of states $i,\ j \in S$, the transition probabilities satisfy the following*

$$p_{ij} > 0, \tag{55}$$

then for all states $j \in S$ the limit

$$\lim_{n\to\infty} p_{ij}(n) = \pi_j, \tag{56}$$

exists and is independent of state $i \in S$.

Proof. See [22].

Theorem 20 *Let $\{X_n\}_{n=0}^{\infty}$ be an Markov chain. If for all pairs of states $i,\ j \in S$, the transition probabilities satisfy the following*

$$p_{ij} > 0, \tag{57}$$

then the limit

$$\lim_{n\to\infty} \pi(n) = \pi, \tag{58}$$

exists for any initial distribution $\pi(0)$.

Proof. Equation 23 gives

$$\pi(n) = \pi(0)P^n,$$

for any initial distribution $\pi(0)$. This can be expressed as

$$P(X_n = j) = \sum_{k=1}^{J} p_{kj}(n)P(X_0 = k). \tag{59}$$

From this it follows

$$\begin{aligned} \lim_{n\to\infty} P(X_n = j) &= \lim_{n\to\infty} \left(\sum_{k=1}^{J} p_{kj}(n)P(X_0 = k) \right) \\ &= \sum_{k=1}^{J} \lim_{n\to\infty} p_{kj}(n)P(X_0 = k). \end{aligned} \tag{60}$$

By assumption 57 and previous theorem equation 60 becomes

$$\begin{aligned} \sum_{k=1}^{J} \lim_{n\to\infty} p_{kj}(n)P(X_0 = k) &= \sum_{k=1}^{J} \pi_j P(X_0 = k) \\ &= \pi_j \sum_{k=1}^{J} \pi_k(0) = 1. \end{aligned}$$

Thus, a sufficient condition for a Markov chain with a finite state space to be ergodic, is the condition $p_{ij} > 0$, for all states $i,\ j \in S$.

Example 5 It is considered the binary information source in Example 1. It is also assumed that the transition probabilities satisfy $p,\ q > 0$. The transition matrix P is clearly positive, thus Theorems 17 and 19 imply, that the invariant distribution π is any row of the following limit

$$\lim_{n\to\infty} P(n).$$

To remind, the transition matrix equals

$$P = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}.$$

By Theorem 10 the (n+1)-step transition matrix becomes

$$\begin{aligned} P(n+1) &= P(n)P \\ &= \begin{pmatrix} p_{11}(n) & p_{12}(n) \\ p_{21}(n) & p_{22}(n) \end{pmatrix} \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix} \\ &= \begin{pmatrix} p_{11}(n)(1-p)+p_{12}(n)q & p_{11}(n)p+p_{12}(n)(1-q) \\ p_{21}(n)(1-p)+p_{22}(n)q & p_{21}(n)p+p_{22}(n)(1-q) \end{pmatrix}. \end{aligned}$$

Since $p_{i1}(n)+p_{i2}(n)=1,\ i=1,\ 2$, it is possible to eliminate $p_{i2}(n),\ i=1,\ 2$, resulting

$$\begin{aligned} P(n+1) &= \begin{pmatrix} p_{11}(n+1) & p_{12}(n+1) \\ p_{21}(n+1) & p_{22}(n+1) \end{pmatrix} \\ &= \begin{pmatrix} p_{11}(n)(1-p-q)+q & p+p_{12}(n)(1-p-q) \\ p_{21}(n)(1-p-q)+q & p+p_{22}(n)(1-p-q) \end{pmatrix}. \end{aligned}$$

Thus, for state 1, it is got two recurrence relations

$$\begin{cases} p_{11}(n+1) = p_{11}(n)(1-p-q)+q \\ p_{12}(n+1) = p+p_{12}(n)(1-p-q) \\ \qquad\qquad = p_{12}(n)(1-p-q)+p \end{cases}$$

It is shown in [33, p. 57], that both of these recurrence relations have unique solutions of the form

$$\begin{cases} p_{11}(n) = \frac{q}{p+q} + A(1-p-q)^n \\ p_{12}(n) = \frac{p}{p+q} + B(1-p-q)^n \end{cases},$$

where A and B are constants. Now, since the sum $p+q<2$, it follows that the absolute value $0<|1-p-q|=|1-(p+q)|<1$. By this we get

$$p_{11}(n) = \frac{q}{p+q} + A(1-p-q)^n \to \frac{q}{p+q},\ n\to\infty,$$

and

$$p_{12}(n) = \frac{p}{p+q} + B(1-p-q)^n \to \frac{p}{p+q},\ n\to\infty.$$

Thus the invariant distribution equals $\pi = (\frac{q}{p+q},\ \frac{p}{p+q})$.

Definition 16 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain. State $j\in S$ is said to be reachable from state $i\in S$, if the n-step transition probability $p_{ij}(n)>0$, for some integer $n\geq 0$.*

Definition 17 *A Markov chain $\{X_n\}_{n=0}^{\infty}$ is said to be irreducible, if for all pairs of states $i,\ j \in S$, the state j is reachable from state i.*

Example 6 We suppose a Markov chain with three states has the transition matrix

$$P = \begin{pmatrix} 0 & 0.7 & 0.3 \\ 0.3 & 0 & 0.7 \\ 1 & 0 & 0 \end{pmatrix}.$$

It is found out, whether this Markov chain is irreducible. According to Definitions 16 and 17, we should find out whether there exist an integer $n \geq 0$ such that for all pairs of states $i,\ j \in S$, the n-step transition probability $p_{ij}(n) > 0$. By Theorem 13 this is done exploring the powers of the transition matrix P:

$$\begin{aligned} P^2 &= \begin{pmatrix} 0.51 & 0 & 0.49 \\ 0.70 & 0.21 & 0.09 \\ 0 & 0.70 & 0.30 \end{pmatrix}, \\ P^3 &= \begin{pmatrix} 0.490 & 0.357 & 0.153 \\ 0.153 & 0.490 & 0.357 \\ 0.510 & 0 & 0.490 \end{pmatrix}, \end{aligned}$$

and

$$P^4 = \begin{pmatrix} 0.2601 & 0.3430 & 0.3969 \\ 0.5040 & 0.1071 & 0.3889 \\ 0.4900 & 0.3570 & 0.1530 \end{pmatrix}.$$

Matrix P^4 is clearly positive and therefore, a transition can be made between any two states in four steps. [1, pp. 224 - 225]

Definition 18 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain. States $j \in S$ and $i \in S$, are said to communicate, if i is reachable from j and j is reachable from i. This is indicated by writing $i \leftrightarrow j$.*

Definition 19 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain. The first return time to the state j is defined by*

$$T_r = \min\{n > 0 |\ X_n = j\}, \tag{61}$$

conditional on, $X_0 = i$.

The probability for the first return time to state j equals

$$f_{jj}(n) = P(X_1 \neq j,\ X_2 \neq j,\ ...\ X_n \neq j|\ X_0 = j). \tag{62}$$

The probability of ever returning to state j is given by

$$f_j = P(X_n = j,\ n > 0|\ X_0 = j) = \sum_{n=1}^{\infty} f_{jj}(n). \tag{63}$$

If $f_j < 1$, then state j is called *transient*. In other words, the chain will eventually leave the transient state without ever returning to it. If $f_j = 1$, then state j is called *recurrent*. The *mean recurrence time* or *the mean return time of state* j is defined as the expected time of return

$$m_j = E(T_r|\ X_0 = j) = \sum_{n=1}^{\infty} n f_{jj}(n). \tag{64}$$

If $m_j = \infty$, then state j is said to be *recurrent null*. If $m_j < \infty$, then state j is said to be *positive recurrent*.

Definition 20 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain. If there exists a state $j \in S$, such that the transition probability equals $p_{jj} = 1$, it is said that state j is absorbing.*

If the Markov chain enters an absorbing state, it never leaves it.

Definition 21 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain and $j \in S$ an absorbing state. The time to absorption is defined as*

$$T_a = \min\{n|\ X_n = j\}. \tag{65}$$

If there is only one absorbing state for the Markov chain, it is possible to renumber the states, so that the absorbing state becomes J. Let the expected value for the time to absorption, on the conditional $X_0 = i$, be denoted by

$$k_i = E(T_a|\ X_0 = i). \tag{66}$$

Shortly said, k_i is referred to as the *expected absorption time*.

Theorem 21 *Let $\{X_n\}_{n=0}^{\infty}$ be a Markov chain with state space $S = \{1,\ 2,\ \ldots\ ,\ J\}$, where state J is the only absorbing state. For each non-absorbing state i, the expected absorption time k_i is an element of the vector*

$$k = (k_1,\ k_2,\ \ldots\ ,\ k_{J-1}),$$

where k is the minimal non-negative solution to the system of linear equations

$$\begin{cases} k_J = 0 \\ k_i = 1 + \sum\limits_{j \neq J} p_{ij} k_j \end{cases} . \tag{67}$$

Proof. See [33, p. 17].

Example 7 It is considered a left-right Markov chain having J states and transition matrix

$$P = \begin{pmatrix} p & q & 0 & \cdots & 0 & 0 \\ 0 & p & q & \cdots & 0 & 0 \\ \vdots & & & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & p & q \\ 0 & 0 & 0 & \cdots & 0 & 1 \end{pmatrix},$$

where $p + q = 1,\ p,\ q > 0$. It is computed the expected time to absorption given, that $X_0 = 1$. State J is here the only absorbing state. It follows from Theorem 21, that the expected absorption time $k_i,\ i \neq J$, is a result of finding the minimal non-negative solution to the following system of linear equations

$$\begin{cases} k_1 = 1 + p_{11}k_1 + p_{12}k_2 + \ldots + p_{1(J-1)}k_{J-1} \\ k_2 = 1 + p_{21}k_1 + p_{22}k_2 + \ldots + p_{2(J-1)}k_{J-1} \\ \vdots \\ k_{J-1} = 1 + p_{(J-1)1}k_1 + p_{(J-1)2}k_2 + \ldots + p_{(J-1)(J-1)}k_{J-1} \end{cases}$$

$$\begin{cases} k_1 = 1 + pk_1 + qk_2 \\ k_2 = 1 + pk_2 + qk_3 \\ \vdots \\ k_{J-2} = 1 + pk_{J-2} + qk_{J-1} \\ k_{J-1} = 1 + pk_{J-1} \end{cases} .$$

Now, solving the last equation results

$$k_{J-1} = \frac{1}{1-p} = \frac{1}{q}.$$

Applying this to the second last equation gives

$$\begin{aligned} k_{J-2} &= 1 + pk_{J-2} + qk_{J-1} \\ &= 1 + pk_{J-2} + 1 \\ &\Leftrightarrow \\ k_{J-2} &= \frac{2}{q}. \end{aligned}$$

Continuing the process gives

$$\begin{aligned} k_{J-(J-1)} &= k_1 = 1 + pk_1 + qk_{J-(J-2)} \\ &= 1 + pk_1 + J - 2 \\ &\Leftrightarrow \\ k_1 &= \frac{J-1}{q}. \end{aligned}$$

Thus the expected time to absorption given, that $X_0 = 1$, equals

$$k_1 = \frac{J-1}{q}.$$

3 Discrete hidden Markov models

3.1 Formal definition and three basic problems

A hidden Markov model (HMM) is a stochastic process generated by two interrelated probabilistic mechanisms. At discrete instants of time, the process is assumed to be in some state and an observation is generated by the random function corresponding to the current state. The underlying Markov chain then changes states according to its probability matrix. The observer sees only the output of the random functions associated with each state and cannot directly observe the states of the underlying Markov chain. This makes the Markov chain hidden. In the following, it is given the more formal definition of an HMM. In the formulation of this chapter, it has been followed the exemplar of [20].

Definition 22 *A hidden Markov model is a parameter triple* $\lambda = (A,\ B,\ \pi(0))$*, with following characterizations:*
(I) (Hidden Markov Chain) A Markov chain $\{X_n\}_{n=0}^{\infty}$ *with a finite state space* $S = \{1,\ 2,\ \ldots\ ,\ J\}$ *, stationary transition matrix*

$$A = (a_{ij})_{i=1,j=1}^{J,J}\ , \tag{68}$$

and initial distribution

$$\pi(0) = (\pi_1(0),\ \pi_2(0),\ \ldots\ \pi_J(0)). \tag{69}$$

(II) (Observable Random Process) A random process $\{Y_n\}_{n=0}^{\infty}$*, with finite state space* $O = (O_1,\ O_2,\ \ldots\ ,\ O_K)$*. The processes* $\{X_n\}_{n=0}^{\infty}$ *and* $\{Y_n\}_{n=0}^{\infty}$ *are related by the following conditional probabilities*

$$b_j(O_k) = P(Y_n = O_k |\ X_n = j),\ n \geq 0. \tag{70}$$

These probabilities are usually presented in the following matrix form

$$B = \{b_j(O_k)\}_{j=1,k=1}^{J,K}\ ,$$

which is called the emission probability matrix. This matrix satisfies the natural stochastic constraints

$$b_j(O_k) \ \geq\ 0, \tag{71}$$

$$\sum_{k=1}^{K} b_j(O_k) \ =\ 1. \tag{72}$$

(III) (Conditional Independence) For any sequence of states $X_0 = j_0,\ X_1 = j_1,\ \ldots\ ,\ X_n = j_n$, *the conditional probability of the observation sequence* $Y = (o_1,\ o_2,\ \ldots\ ,\ o_n)$, $o_l \in O$, $l = 1,\ 2,\ \ldots\ ,\ n$, *equals*

$$\begin{aligned} P(Y_0 &= o_0,\, Y_1 = o_1,\, \ldots\,,\, Y_n = o_n |\, X_0 = j_0,\ X_1 = j_1,\ \ldots\ ,\ X_n = j_n,\ \lambda) \\ &= \prod_{l=0}^{n} b_{j_l}(o_l). \end{aligned} \tag{73}$$

It should be noted here, that with the notation o_l, it is meant an observation at time instant l, where $o_l \in O$.

Theorem 22 *Let the triple* $\lambda = (A,\ B,\ \pi(0))$ *be an HMM. For an observation sequence* $Y = (o_1,\ o_2,\ \ldots\ ,\ o_n)$ *the conditional probability equals*

$$P(O|\ \lambda) = \sum_{j_0=1}^{n} \cdots \sum_{j_n=1}^{n} P(Y,\ X|\ \lambda),$$

where

$$\begin{aligned} & P(Y,\, X|\, \lambda) \\ = \ & \pi_{j_0}(0) \prod_{l=0}^{n} b_{j_l}(o_l) \prod_{l=1}^{n} a_{j_{l-1} j_l}, \end{aligned}$$

and $X = (X_0 = j_0,\, X_1 = j_1,\, \ldots\,,\, X_n = j_n)$ *is a state sequence.*

Proof. By Theorems 2 and 7, and condition 73 the joint probability becomes

$$\begin{aligned} P(Y,\, X|\, \lambda) &= P(Y|\, X,\, \lambda) P(X|\, \lambda) \\ &= \prod_{l=0}^{n} b_{j_l}(o_l)\, \pi_{j_0}(0) \prod_{l=1}^{n} a_{j_{l-1} j_l}. \end{aligned}$$

By rearrangement the last expression becomes

$$P(Y,\ X|\ \lambda) = \pi_{j_0}(0)\, b_{j_0}(o_0) \prod_{l=1}^{n} a_{j_{l-1} j_l}\, b_{j_l}(o_l).$$

Now, by summing over all possible paths of state sequence the joint probability becomes

$$\begin{aligned} P(Y|\ \lambda) &= \sum_{j_0=1}^{J} \cdots \sum_{j_n=1}^{J} P(Y,\ X|\ \lambda) \\ &= \sum_{j_0=1}^{J} \cdots \sum_{j_n=1}^{J} \pi_{j_0}(0)\, b_{j_0}(o_0) \prod_{l=1}^{n} a_{j_{l-1} j_l}\, b_{j_l}(o_l). \end{aligned}$$

It is possible to see an HMM as a "probabilistic mechanism" generating observation sequences. The algorithm for this generation process, using random number generator whose output is uniform on the interval $[0, 1]$, can be found in [25]. This algorithm is now presented here assuming an HMM with parameter set $\lambda = (A,\ B,\ \pi(0))$:

HMM Generator of Observations

(1) Partition the unit interval proportionally to the components of the initial distribution $\pi(0)$. Generate a random number and select a start state, X_0, according to the subinterval in which the number falls. Set time $t = 1$.
(2) Partition the unit interval proportionally to the components of ith row of the emission probability matrix B. Generate a random number and select an observation, o_t, according to the subinterval in which the number falls.
(3) Partition the unit interval proportionally to the components of ith row of the transition matrix A. Generate a random number and select the next state, X_t, according to the subinterval in which the number falls.
(4) Increment time t by unit. If $t \leq T$, repeat steps (2)-(4), otherwise stop.

As a result of this algorithm an observation sequence $Y = (o_1,\ o_2,\ ...\ ,\ o_T)$, assuming an HMM $\lambda = (A,\ B,\ \pi(0))$, is generated.

For an HMM to be used in real-world applications, there exist three basic problems of interest to be solved. These problems are introduced according to the presentations in, for example, [36]:

(1) *The Evaluation Problem.*

Given the observation sequence $Y = (o_1,\ o_2,\ \ldots,\ o_T)$ and model $\lambda = (A,\ B,$
$\pi(0))$, it is to compute the probability of the observation sequence $P(Y|\ \lambda)$.

(2) *The Decoding Problem.*
Given the observation sequence $Y = (o_1,\ o_2,\ \ldots,\ o_T)$ and the model $\lambda = (A,\ B,\ \pi(0))$, it is to find a state sequence $X = (X_1 = j_1,\ \ldots,\ X_T = j_T)$, that is optimal in some meaningful sense.

(3) *The Estimation Problem or Training Problem.*
Given the observation sequence $Y = (o_1,\ o_2,\ \ldots,\ o_T)$, it is to find the parameters $\lambda = (A,\ B,\ \pi(0))$ to maximize the probability $P(Y|\ \lambda)$.

The evaluation problem is already solved straightforwardly, according to Theorem 22. Since the summation is performed over J^{n+1} possible sequences, the total complexity becomes of the order $2(n+1)J^{n+1}$ operations. This makes an overwhelming problem as the length of the observation sequence grows. For this problem, it is presented the *forward-backward procedure* as a reasonable solution in the sequel.

There exist several ways to define the criteria of optimality in the decoding problem. In practice, the most often used criteria, is to find the state sequence $j_0^*,\ j_1^*,\ \ldots,\ j_n^*$, that maximizes the probability

$$P(X_0 = j_0,\ X_1 = j_1,\ \ldots,\ X_n = j_n,\ Y_0 = o_0,\ Y_1 = o_1,\ \ldots,\ Y_n = o_n|\ \lambda),$$

for a given observation sequence $Y = (o_1,\ o_2,\ \ldots,\ o_T)$. This is due to the fact that, this calculation can be implemented by the *Viterbi algorithm*, presented in the sequel.

There exist several ways of solving the estimation problem, also:

(1) *Maximum likelihood method.* In this method it is defined the optimal model parameters as follows

$$\hat{\lambda}_{ML} = \arg\max_{\lambda} P(Y|\ \lambda). \tag{74}$$

The computation of $\hat{\lambda}_{ML}$ is done by using the *Baum-Welch algorithm*, which is of fundamental and historical importance in the application of

HMMs. [7]

(2) *Minimum discrimination information method.* This method is based on minimization of a suitable Kullback distance and is developed in [13].

(3)*Smooth learning algorithms.* This technique is developed in [3] and makes use of the gradient descent.

(4)*Viterbi training*. In this method it is tried to find the model parameters λ so as to maximize the parameter

$$V^*(\lambda) = \max_{all\ state\ sequences\ X} P(X,\ Y|\ \lambda),$$

which is known from the problem of finding the optimal state sequence. This method has been developed in [17] under the name of *segmental k-means algorithm.*

3.2 Auxiliary properties of the conditional distributions

For the exact mathematical analysis of the three basic problems of the hidden Markov modeling, it is presented, in the following, additional properties of the conditional distributions present in HMM. These properties are given as theorems, for which the proofs are available in [28, pp. 202 - 206]. For simplicity, it is used the following notation

$$\begin{aligned} P(Y_m &= o_m,\ \ldots,\ Y_N = o_N|\ X_n = j_n,\ \ldots,\ X_N = j_N) \\ &= P(Y_m,\ \ldots,\ Y_N|\ X_n,\ \ldots,\ X_N). \end{aligned}$$

It is assumed silently the HMM $\lambda = (A,\ B,\ \pi(0))$, which is omitted in the expressions. The following three theorems show that the probability of a finite length observation sequence emitted from an HMM, conditional on a state sequence, depends only on a subsequence of the state sequence.

Theorem 23 *For all integers n and m, $0 \leq n \leq m \leq N$, the conditional probability equals*

$$P(Y_m,\ \ldots,\ Y_N|\ X_n,\ \ldots,\ X_N) = P(Y_m,\ \ldots,\ Y_N|\ X_m,\ \ldots,\ X_N).$$

Theorem 24 *For all integers* $n = 0, \ \ldots \ , \ N-1$ *the conditional probability equals*

$$P(Y_{n+1}, \ \ldots \ , \ Y_N | \ X_0, \ \ldots \ , \ X_N) = P(Y_{n+1}, \ \ldots \ , \ Y_N | \ X_n).$$

Theorem 25 *For all integers* $n = 0, \ \ldots \ , \ N$ *the conditional probability equals*

$$P(Y_0, \ \ldots \ , \ Y_n | \ X_0, \ \ldots \ , \ X_N) = P(Y_0, \ \ldots \ , \ Y_n | \ X_0, \ \ldots \ , \ X_n).$$

The following three factorization theorems highlight the renewal properties of the HMM.

Theorem 26 *For all integers* $n = 0, \ \ldots \ , \ N$ *the conditional probability equals*

$$P(Y_0, \ \ldots \ , \ Y_n | \ X_n) = P(Y_0, \ \ldots \ , \ Y_n | \ X_n) P(Y_{n+1}, \ \ldots \ , \ Y_N | \ X_n).$$

The conditional probability $P(Y_{n+1}, \ \ldots \ , \ Y_N | \ X_n)$ is called the *backward variable,* which is the probability of the emitted subsequence from time $n+1$ to the end N, on the condition, that the hidden Markov chain is in state X_n at time n. Next two theorems are essential to find the recursion for the computation of the backward variable.

Theorem 27 *For all integers* $n = 0, \ \ldots \ , \ N$ *the conditional probability equals*

$$P(Y_n, \ \ldots \ , \ Y_N | \ X_n) = P(Y_n | \ X_n) P(Y_{n+1}, \ \ldots \ , \ Y_N | \ X_n).$$

Theorem 28 *For all integers* n *and* $m, \ 0 \leq n \leq m \leq N$, *the conditional probability equals*

$$P(Y_m, \ \ldots \ , \ Y_N | \ X_n, \ \ldots \ , \ X_m) = P(Y_m, \ \ldots \ , \ Y_N | \ X_m).$$

The conditional probability $P(Y_0, \ \ldots \ , \ Y_n, \ X_n)$ is called the *forward variable,* which is the simultaneous probability of the emitted subsequence up to time $n \leq N$ and of the hidden Markov chain to be in state X_n at time n. The last theorem is useful in finding the properties of this variable.

Theorem 29 *For all integers* $n = 0, \ \ldots \ , \ N-1$ *the conditional probability equals*

$$P(Y_0, \ \ldots \ , \ Y_N | \ X_n, \ X_{n+1}) = P(Y_0, \ \ldots \ , \ Y_N | \ X_n) P(Y_{n+1}, \ \ldots \ , \ Y_N | \ X_{n+1}).$$

3.3 Evaluation problem and forward-backward algorithm

The fundamental work [7] of Baum and Petrie created the computational possibilities for the effective applications of HMMs. In [7] it is presented a computationally effective solution to the evaluation or *scoring* problem for HMM. This algorithm is called the *forward-backward algorithm,* which in formal sense, is derived from the seven previous theorems presented. It is again assumed silently the HMM $\lambda = (A,\ B,\ \pi(0))$, which is omitted in the expressions. The evolution of the forward-backward algorithm is based on the utilization of the forward and backward variables defined previously. In the sequel, the forward variable is denoted by

$$\alpha_n(j) = P(Y_0 = o_0,\, ...\,,\, Y_n = o_n,\, X_n = j),\ 0 \le n \le N$$

and the backward variable is denoted by

$$\beta_n(j) = P(Y_{n+1} = o_{n+1},\, ...\,,\, Y_N = o_N \,|\, X_n = j),\ 0 \le n \le N.$$

For convenience, it is chosen $\beta_N(j) = 1,\ \forall\, j.$

Theorem 30 *For an observation sequence* $Y = (Y_0 = o_0,\, ...\,,\, Y_N = o_N)$*, the joint probability equals*

$$P(Y_0 = o_0,\ ...\ ,\ Y_n = o_n) = \sum_{j=1}^{J} \alpha_n(j)\beta_n(j).$$

Proof. By Theorem 2, the conditional probability equals

$$\begin{aligned} & P(Y_0 = o_0,\ ...\ ,\ Y_n = o_n,\ X_n = j) \\ = \ & P(X_n = j)P(Y_0 = o_0,\ ...\ ,\ Y_n = o_n |\ X_n = j). \end{aligned} \tag{75}$$

By Theorem 26 the right hand side this equation is factorized as follows

$$\begin{aligned} & P(X_n = j)P(Y_0 = o_0,\ ...\ ,\ Y_n = o_n |\ X_n = j) \\ = \ & P(X_n = j)P(Y_0 = o_0,\ ...\ ,\ Y_n = o_n |\ X_n = j) \cdot \\ & \quad \cdot P(Y_{n+1} = o_{n+1},\ ...\ ,\ Y_N = o_N |\ X_n = j) \\ = \ & P(Y_0 = o_0,\ ...\ ,\ Y_n = o_n,\ X_n = j) \cdot \\ & \quad \cdot P(Y_{n+1} = o_{n+1},\ ...\ ,\ Y_N = o_N |\ X_n = j) \\ = \ & \alpha_n(j)\beta_n(j). \end{aligned} \tag{76}$$

Now, we get

$$\begin{aligned} P(Y_0 = o_0,\ \ldots\ ,\ Y_n = o_N) &= \sum_{j=1}^{J} P(Y_0 = o_0,\ \ldots\ ,\ Y_n = o_n,\ X_n = j) \\ &= \sum_{j=1}^{J} \alpha_n(j)\beta_n(j). \end{aligned}$$

Theorem 31 *Let $Y = (Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N)$, be an observation sequence produced by an HMM. The forward variables $\alpha_n(j) = P(Y_0 = o_0,\ \ldots\ ,\ Y_n = o_n,\ X_n = j),\ 0 \le n \le N,\ j = 1,\ \ldots\ ,\ J$, result from the following recursion process:*

Initialization:

$$\alpha_0(j) = \pi_j(0)b_j(o_0),\ j = 1,\ \ldots\ , J.$$

Recursion:

$$\alpha_{n+1}(j) = \left[\sum_{i=1}^{J} \alpha_n(i)a_{ij}\right] b_j(o_{n+1}),\ 0 \le n \le N,\ j = 1,\ \ldots\ ,\ J. \quad (77)$$

Proof. By definition 2 the forward probability equals

$$\begin{aligned} \alpha_0(j) &= P(Y_0 = o_0,\ X_0 = j) = P(Y_0 = o_0 |\ X_0 = j)P(X_0 = j) \\ &= \pi_j(0)b_j(o_0), \end{aligned}$$

$j = 1, \ldots, J$. This makes the initialization sensible. Let the state $j \in \{1,\ \ldots\ ,\ J\}$, and $n \ge 0$, be arbitrary. By Theorem 29 and Definition 2, the forward variable becomes

$$\begin{aligned} \alpha_{n+1}(j) &= P(Y_0 = o_0,\ \ldots\ ,\ Y_{n+1} = o_{n+1},\ X_{n+1} = j) \\ &= \sum_{i=1}^{J} P(Y_0 = o_0,\ \ldots\ ,\ Y_{n+1} = o_{n+1},\ X_n = i,\ X_{n+1} = j) \\ &= \sum_{i=1}^{J} P(X_n = i,\ X_{n+1} = j) \cdot \\ &\quad \cdot P(Y_0 = o_0,\ \ldots\ ,\ Y_{n+1} = o_{n+1} |\ X_n = i,\ X_{n+1} = j) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{J} P(X_n = i,\ X_{n+1} = j) \cdot \\
&\quad \cdot P(Y_0 = o_0,\ \ldots,\ Y_n = o_n |\ X_n = i) \cdot \\
&\quad \cdot P(Y_{n+1} = o_{n+1} |\ X_{n+1} = j). \qquad (78)
\end{aligned}
$$

Again, by the definition of conditional probability, we get

$$
\begin{aligned}
P(X_n &= i,\ X_{n+1} = j) P(Y_{n+1} = o_{n+1} |\ X_{n+1} = j) \\
&= P(X_{n+1} = j |\ X_n = i) P(X_n = i) P(Y_{n+1} = o_{n+1} |\ X_{n+1} = j) \\
&= a_{ij} b_j(o_{n+1}) P(X_n = i). \qquad (79)
\end{aligned}
$$

Since

$$
\begin{aligned}
P(Y_0 &= o_0,\ \ldots,\ Y_n = o_n |\ X_n = i) P(X_n = i) \\
&= P(Y_0 = o_0,\ \ldots,\ Y_n = o_n,\ X_n = i) \\
&= \alpha_n(j),
\end{aligned}
$$

then by equations 78 and 79 we get

$$
\begin{aligned}
\alpha_{n+1}(j) &= \sum_{i=1}^{J} P(X_n = i,\ X_{n+1} = j) P(Y_0 = o_0,\ \ldots\ ,\ Y_n = o_n |\ X_n = i) \cdot \\
&\qquad \cdot P(Y_{n+1} = o_{n+1} |\ X_{n+1} = j) \\
&= \sum_{i=1}^{J} \alpha_n(j) a_{ij} b_j(o_{n+1}) \\
&= \left[\sum_{i=1}^{J} \alpha_n(j) a_{ij} \right] b_j(o_{n+1}).
\end{aligned}
$$

A trellis structure for the recursive calculation of the forward variables is presented in Figure 3. [37]

Theorem 32 *Let $Y = (Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N)$, be an observation sequence produced by an HMM. The backward variables $\beta_n(j) = P(Y_{n+1} = o_{n+1},\ \ldots\ ,\ Y_N = o_N |\ X_n = j),\ 0 \le n \le N,\ j = 1,\ \ldots\ ,\ J$, result from the following recursion process:*

Initialization:

$$\beta_N(j) = 1,\ j = 1,\ \ldots\ , J.$$

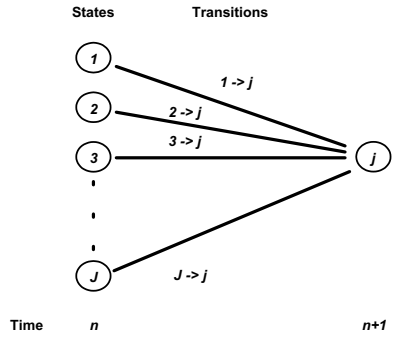

Figure 3. Illustration of the sequence of operations needed for the computation of the forward variables.

Recursion:

$$\beta_n(j) = \sum_{i=1}^{J} b_i(o_{n+1})\beta_{n+1}(i)a_{ji},\ 0 \le n \le N,\ j = 1,\ \ldots\ ,\ J, \qquad (80)$$

where $n = N-1,\ N-2,\ \ldots\ ,\ 0$, *and* $j = 1,\ \ldots\ ,\ J$.

Proof. Let $n \in \{N-1,\ N-2,\ \ldots\ ,\ 0\}$, and $j \in \{1,\ \ldots\ ,\ J\}$ be arbitrary. By Definition 2 and Theorem 2 the backward probability becomes

$$\begin{aligned} & \beta_n(j) = P(Y_{n+1} = o_{n+1},\ \ldots\ ,\ Y_N = o_N |\ X_n = j) \\ = & \sum_{i=1}^{J} P(Y_{n+1} = o_{n+1},\ \ldots\ ,\ Y_N = o_N,\ X_{n+1} = i |\ X_n = j) \\ = & \sum_{i=1}^{J} \frac{P(Y_{n+1} = o_{n+1},\ \ldots\ ,\ Y_N = o_N |\ X_n = j,\ X_{n+1} = i)}{P(X_n = j)} \cdot \\ & \cdot P(X_n = j,\ X_{n+1} = i). \end{aligned}$$

By Theorems 28 and 27 the conditional probability becomes

$$\begin{aligned} P(Y_{n+1} &= o_{n+1},\ \ldots\ ,\ Y_N = o_N |\ X_n = j,\ X_{n+1} = i) \\ &= P(Y_{n+1} = o_{n+1},\ \ldots\ ,\ Y_N = o_N |\ X_{n+1} = i) \\ &= P(Y_{n+1} = o_{n+1} |\ X_{n+1} = i) \cdot \\ &\quad \cdot P(Y_{n+2} = o_{n+2},\ \ldots,\ Y_N = o_N |\ X_{n+1} = i). \end{aligned} \qquad (81)$$

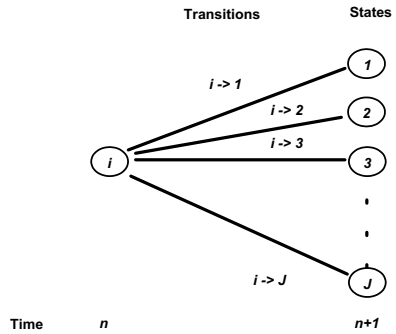

Figure 4. Illustration of the sequence of the operations needed for the computation of the backward variable.

It was stated in the previous theorem, that the transition probability equals

$$a_{ji} = \frac{P(X_n = j,\ X_{n+1} = i)}{P(X_n = j)}.$$

Now, by this and equation 81, the backward variable becomes

$$\begin{aligned} \beta_n(j) &= \sum_{i=1}^{J} P(Y_{n+1} = o_{n+1} |\ X_{n+1} = i) \cdot \\ &\quad \cdot P(Y_{n+2} = o_{n+2},\ \ldots\ ,\ Y_N = o_N |\ X_{n+1} = i) a_{ji} \\ &= \sum_{i=1}^{J} b_i(o_{n+1}) \beta_{n+1}(i) a_{ji}. \end{aligned}$$

A trellis structure for the recursive calculation of the backward variables is presented in Figure 4. [37]

Both forward and backward procedures are clearly of complexity J^2T, which shows, that the needed amount of calculations grows linearly with the number of observations. Forward and backward procedures offer an effective way of solving the evaluation problem according to Theorem 30 as follows

$$P(Y_0 = o_0,\ \ldots\ ,\ Y_n = o_n) = \sum_{j=1}^{J} \alpha_n(j) \beta_n(j).$$

This holds for arbitrary $n \in \{0,\ 1,\ \ldots\ ,\ N\}$. In speech recognition, the often used solution for the evaluation problem is choosing $n = N$:

$$P(Y_0 = o_0,\ \ldots\ ,\ Y_n = o_n) = \sum_{j=1}^{J} \alpha_N(j)\beta_N(j)$$

$$= \sum_{j=1}^{J} \alpha_N(j),$$

by the convention $\beta_N(j) = 1$. [37]

3.4 Decoding with smoothing and the Viterbi algorithm

Unlike the scoring problem, there exist several possible ways of solving the problem of finding the "optimal" state sequence associated with the given observation sequence Y. It is possible to define the "optimality" criteria in many ways. The smoothing method chooses the states that are individually most likely at each time instant, while the Viterbi algorithm tries to find the single best state sequence X to maximize the likelihood $P(X|\ Y)$. The latter method is based on dynamic programming methods and it was first presented in [48].

Definition 23 *Let $Y = (Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N)$, be an observation sequence produced by an HMM. The smoothing probability $\overset{\wedge}{\pi}\ (n|\ N),\ n \leq N$, is defined by*

$$\overset{\wedge}{\pi}_j\ (n|\ N) = P(X_N = j|\ Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N), \tag{82}$$

where $j = 1,\ 2,\ \ldots\ ,\ J$ and $0 \leq n \leq N$.

The smoothing probability can be seen as an *a posteriori* variable containing all probabilistic information the state X_n, given an observation sequence $Y = (Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N)$.

Theorem 33 *Let $Y = (Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N)$, be an observation sequence produced by an HMM. Thus, the smoothing probability equals*

$$\overset{\wedge}{\pi}_j\ (n|\ N) = \frac{\alpha_n(j)\beta_n(j)}{\sum_{j=1}^{J} \alpha_n(j)\beta_n(j)},$$

where $\alpha_n(j)$ and $\beta_n(j)$ are the forward and backward variables corresponding to the observation sequence Y.

Proof. By the definition of the conditional probability, the smoothing probability becomes

$$\begin{aligned}
\overset{\wedge}{\pi}_j\ (n|\ N) &= P(X_n = j|\ Y_0 = o_0,\ \ldots,\ Y_N = o_N) \\
&= \frac{P(Y_0 = o_0,\ \ldots,\ Y_N = o_N,\ X_n = j)}{P(Y_0 = o_0,\ \ldots,\ Y_N = o_N)} \\
&= \frac{P(Y_0 = o_0,\ \ldots,\ Y_N = o_N,\ X_N = j)}{P(Y_0 = o_0,\ \ldots,\ Y_N = o_N)}.
\end{aligned}$$

By equations 75 and 76, the joint probability equals

$$P(Y_0 = o_0,\ \ldots,\ Y_N = o_N,\ X_N = j) = \alpha_n(j)\beta_n(j).$$

Now, by Theorem 30, the smoothing becomes

$$\begin{aligned}
\overset{\wedge}{\pi}_j\ (n|\ N) &= \frac{P(Y_0 = o_0,\ \ldots,\ Y_N = o_N,\ X_N = j)}{P(Y_0 = o_0,\ \ldots,\ Y_N = o_N)} \\
&= \frac{\alpha_n(j)\beta_n(j)}{\sum\limits_{j=1}^{J} \alpha_n(j)\beta_n(j)}.
\end{aligned}$$

Given an observation sequence $Y = (Y_0 = o_0,\ \ldots,\ Y_N = o_N)$ and an HMM, the decoding problem could be solved, by finding the optimal state $j^*(n)$, such that

$$j^*(n) = \arg \max_{1 \le j \le J} \overset{\wedge}{\pi}\ (n|\ N), \tag{83}$$

for all time instants $n = 0,\ 1,\ \ldots\ ,\ N$. The problem, that this solution produces, is since the rule works by taking into account the state at an instant. This can produce a nonvalid state sequence, if there are present forbidden state transitions in the hidden Markov chain.

Another way in solving the decoding or alignment problem, is to find the state sequence $X = (X_0 = j_0,\ \ldots,\ X_N = j_N)$ that maximizes the joint probability

$$P(Y_0 = o_0,\ \ldots,\ Y_N = o_N,\ X_0 = j_0,\ \ldots,\ X_N = j_N),$$

given the observation sequence $Y = (Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N)$ and an HMM. For an partial observation sequence and a subsequence of states it is used following notations

$$Y^{(n)} = (Y_0 = o_0,\ \ldots\ ,\ Y_n = o_n),$$

and

$$X^{(n)} = (X_0 = j_0,\ \ldots\ ,\ X_n = j_n).$$

Given an observation sequence $Y = (Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N)$ and an HMM, the *best score variable* $\delta_n(j)$ is defined by

$$\delta_n(j) = \max_{j_0,\ldots,j_{n-1}} P(Y^{(n)},\ X_0 = j_0,\ \ldots\ ,\ X_n = j),\ 0 \le n \le N.$$

Let the subsequence $X^{(n)}$ of states ending at state j be denoted by $X_j^{(n)}$. The best score variable $\delta_n(j)$ is the highest joint probability of a subsequence $X_j^{(n)}$ of states, and corresponding partial observation sequence $Y^{(n)}$.

Theorem 34 (Bellman's Optimality Principle) *Let $Y = (Y_0 = o_0,\ \ldots\ , Y_N = o_N)$, be an observation sequence produced by an HMM. The best score variables $\delta_n(j),\ 0 \le n \le N,\ j = 1,\ \ldots\ ,\ J$, result from the following recursion process:*

Initialization:

$$\delta_0(j) = \pi_j(0) b_j(o_0)$$

Recursion:

$$\delta_n(j) = \left[\max_{i=1,\ldots,J} \delta_{n-1}(i) a_{ij}\right] b_j(o_n),$$

where $n = 1,\ \ldots\ ,\ N$, and $j = 1,\ \ldots\ ,\ J$.

Proof. Let integer $n = 0$ and state $j \in \{1,\ \ldots\ ,\ J\}$ be arbitrary. By the definitions of conditional probability we get

$$\begin{aligned} \delta_0(j) &= P(Y_0 = o_0,\ X_0 = j) \\ &= P(Y_0 = o_0 |\ X_0 = j) P(X_0 = j) \\ &= \pi_j(0) b_j(o_0). \end{aligned}$$

Now, by the definition of conditional probability, conditional independence, and the Markov property the joint probability becomes

$$\begin{aligned}
& P(Y^{(n)},\ X_j^{(n)}) \\
= \ & P(Y^{(n)}|\ X_j^{(n)})P(X_j^{(n)}) \\
= \ & P(Y_n|\ X_n = j)\prod_{k=1}^{n-1} P(Y_k|\ X_k)P(X_n = j|\ X^{(n-1)})P(X^{(n-1)}) \\
= \ & P(Y_n|\ X_n = j)P(X_n = j|\ X_{n-1})\prod_{k=1}^{n-1} P(Y_k|\ X_k)P(X^{(n-1)}) \\
= \ & b_j(o_n)a_{j_{n-1},j}\,P(Y^{(n-1)}|\ X^{(n-1)})P(X^{(n-1)}) \\
= \ & b_j(o_n)a_{j_{n-1},j}\,P(Y^{(n-1)},\ X^{(n-1)}).
\end{aligned}$$

By this and the definition of the score variable we get

$$\begin{aligned}
\delta_n(j) &= \max_{j_0,\ldots,j_{n-1}} P(Y^{(n)},\ X_0 = j_0,\ \ldots\ ,\ X_n = j) \\
&= b_j(o_n)\max_{j_0,\ldots,j_{n-1}} a_{j_{n-1},j}\,P(Y^{(n-1)},\ X^{(n-1)}). \qquad (84)
\end{aligned}$$

For each state $X_n = j$, it is to find the transition from every state $X_{n-1} = i \in S$, to maximize the product

$$\begin{aligned}
& a_{j_{n-1},j}\,P(Y^{(n-1)},\ X^{(n-1)}) \\
= \ & a_{j_{n-1},j}\,P(Y^{(n-1)},\ X_0 = j_0,\ \ldots\ ,\ X_{n-1} = j_{n-1}).
\end{aligned}$$

It is possible, that there exist several possible paths of states to state $X_{n-1} = i$. To do the maximization, for each state $X_{n-1} = i$, the special subsequence of states leading to state i giving maximum to the probability $P(Y^{(n-1)},\ X_0 = j_0,\ \ldots\ ,\ X_{n-1} = i)$, has to be chosen. This results the best score variable $\delta_{n-1}(i)$. Now, by equation 84 the best score variable becomes

$$\begin{aligned}
\delta_n(j) &= b_j(o_{n+1})\max_{j_0,\ldots,j_{n-1}} a_{j_{n-1},j}\,P(Y^{(n-1)},\ X^{(n-1)}) \\
&= \left[\max_{j=1,\ldots,J} a_{ij}\delta_{n-1}(i)\right] b_j(o_n).
\end{aligned}$$

For each state the j at time n, the state at time $n-1$, giving the maximum probability $\max_{i=1,\ldots,J} a_{ij}\delta_{n-1}(i)$ is denoted by $\psi_n(j)$. To get the optimal state sequence that maximizes the joint probability

$$P(Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N,\ X_0 = j_0,\ \ldots\ ,\ X_N = j_N),$$

at each time instant $n = 0,\ 1,\ \ldots\ ,\ N$, variable $\psi_{n-1}(j) = j^*_{n-1}$ is recorded. At the last stage, the state giving the biggest best score variable is the last state in the optimal state sequence.

By the previous theorem the complete procedure, called the Viterbi algorithm, for solving the decoding problem can be formalized as follows:

The Viterbi Algorithm

Initialization:

$$\begin{aligned} \delta_0(j) &= \pi_j(0)b_j(o_0), \\ \psi_0(j) &= 0, \end{aligned}$$

where $j = 1,\ \ldots\ ,\ J$.

Recursion:

$$\begin{aligned} \delta_n(j) &= \left[\max_{i=1,\ldots,J} \delta_{n-1}(i)a_{ij}\right] b_j(o_n), \\ \psi_n(j) &= \arg\max_{i=1,\ldots,J} a_{ij}\delta_{n-1}(i), \end{aligned}$$

where $n = 0,\ \ldots\ ,\ N$, and $j = 1,\ \ldots\ ,\ J$.

Termination:

$$\begin{aligned} P^* &= \max_{i=1,\ldots,J} \delta_N(i), \\ j^*(N) &= \arg\max_{i=1,\ldots,J} \delta_N(i). \end{aligned}$$

Best Path Construction:

$$j^*(n) = \psi_{n+1}(j^*(n+1)),\ n = N-1,\ N-2,\ \ldots\ ,0.$$

The Viterbi algorithm is similar, despite the backtracking steps, in implementation to the forward-backward algorithms. Though, a maximization over previous states is used in place of the summing procedure. By taking logarithms of the model parameters, the Viterbi algorithm can be implemented without need for any multiplications:

The Log-Viterbi Algorithm

Preprocessing:

$$\begin{aligned} \widetilde{\pi}_j\,(0) &= \log \pi_j(0), \\ \widetilde{b}_j\,(o_n) &= \log b_j(o_n), \\ \widetilde{a}_{ij} &= \log a_{ij}, \end{aligned}$$

where $n = 0,\ \ldots\ ,\ N$, and $i,\ j = 1,\ \ldots\ ,\ J$.

Initialization:

$$\begin{aligned} \widetilde{\delta}_0\,(j) &= \widetilde{\pi}_j\,(0) + \widetilde{b}_j\,(o_0), \\ \widetilde{\psi}_0\,(j) &= 0, \end{aligned}$$

where $j = 1,\ \ldots\ ,\ J$.

Recursion:

$$\begin{aligned} \widetilde{\delta}_{n+1}\,(j) &= \left[\max_{i=1,\ldots,J} \widetilde{\delta}_{n-1}\,(i) + \widetilde{a}_{ij}\right] + \widetilde{b}_j\,(o_{n+1}), \\ \widetilde{\psi}_n\,(j) &= \arg \max_{i=1,\ldots,J} \left[\widetilde{\delta}_{n-1}\,(i) + \widetilde{a}_{ij}\right], \end{aligned}$$

where $n = 0,\ \ldots\ ,\ N$, and $i,\ j = 1,\ \ldots\ ,\ J$.

Termination:

$$\begin{aligned} \widetilde{P}^* &= \max_{i=1,\ldots,J} \widetilde{\delta}_N\,(i), \\ j^*(N) &= \arg \max_{i=1,\ldots,J} \widetilde{\delta}_N\,(i). \end{aligned}$$

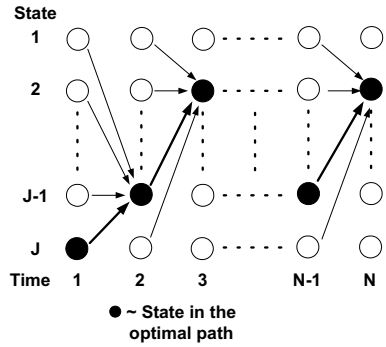

Figure 5. Trellis diagram for the Viterbi algorithm.

Best Path Construction:

$$j^*(n) = \psi_{n+1}(j^*(n+1)),\ n = N-1,\ N-2,\ \ldots\ ,\ 1.$$

Previous algorithm follows the presentation in [38, p. 340]. The calculation for this alternative implementation is on the order of N^2T additions added to calculations needed for the preprocessing. The preprocessing needs to be performed once and saved, thus its cost is odd for most systems. The computation is efficiently implemented by a trellis structure, which is illustrated in Figure 5. [38, p. 340]

The log-Viterbi algorithm handles automatically the problem of small counts, namely the small probabilities. The same problem is faced within the forward and backward algorithms. In the following example, the needed algorithms are implemented using MATLAB.

Example 8 It is given two discrete 3-state HMMs $\lambda_1 = (A_1,\ B_1,\ \pi_1(0))$ and $\lambda_2 = (A_2,\ B_2,\ \pi_2(0))$, with the following stochastic parameters:

$$A_1 = \begin{pmatrix} 0.3 & 0.7 & 0 \\ 0 & 0.6 & 0.4 \\ 0 & 0 & 1 \end{pmatrix},\ A_2 = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.3 & 0.7 \\ 0 & 0 & 1 \end{pmatrix},$$

$$B_1 = \begin{pmatrix} 0.1 & 0.4 & 0.7 \\ 0.9 & 0.6 & 0.3 \end{pmatrix},\ B_2 = \begin{pmatrix} 0.8 & 0.35 & 0.1 \\ 0.2 & 0.65 & 0.9 \end{pmatrix},$$

and

$$\pi_1(0) = \begin{pmatrix} 1,\ 0,\ 0 \end{pmatrix},\ \pi_2(0) = \begin{pmatrix} 1,\ 0,\ 0 \end{pmatrix}.$$

Let the observable process have the state space $O = \{1,\ 2\}$. These HMMs are called left-right HMMs due to the left-right topology of the underlying Markov chain. Using the HMM generator of observations algorithm, it is now generated two observation sequences Y_{λ_1} and Y_{λ_2} of 20 observations:

$$\begin{aligned} Y_{\lambda_1} &= (2,\ 2,\ 2,\ 1,\ 1,\ 2,\ 2,\ 1,\ 1,\ 1) \\ Y_{\lambda_2} &= (1,\ 1,\ 2,\ 2,\ 2,\ 1,\ 1,\ 2,\ 2,\ 2) \end{aligned}$$

During the generation process, it is possible to record the "real" state sequences corresponding to the observation sequences:

$$\begin{aligned} X_{\lambda_1} &= (1,\ 1,\ 1,\ 2,\ 2,\ 3,\ 3,\ 3,3,\ 3) \\ X_{\lambda_2} &= (1,\ 2,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3) \end{aligned}$$

As a result of the forward algorithm, we get conditional probabilities, that give a measure for a given model to produce a given observation sequence. Due to small numerical value of these probabilities, it is more often presented the corresponding log-probabilities,

$$\begin{aligned} \log P(Y_{\lambda_1}|\,\lambda_1) &= -5.4647, \\ \log P(Y_{\lambda_1}|\,\lambda_2) &= -11.8636, \\ \log P(Y_{\lambda_2}|\,\lambda_1) &= -7.8953, \end{aligned}$$

and

$$\log P(Y_{\lambda_2}|\,\lambda_2) = -6.3437.$$

Using the log-Viterbi algorithm, it is possible to get the most probable state sequences and corresponding log-probabilities for given observation sequence and model:

$$\begin{aligned} \log P(Y_{\lambda_1},\, X^*_{Y_{\lambda_1},\lambda_1}|\,\lambda_1) &= -7.1021, \\ X^*_{Y_{\lambda_1},\lambda_1} &= (1,\ 2,\ 2,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3), \\ \log P(Y_{\lambda_1},\, X^*_{Y_{\lambda_1},\lambda_2}|\,\lambda_2) &= -14.9190, \\ X^*_{Y_{\lambda_1},\lambda_2} &= (1,\ 2,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3) \\ \log P(Y_{\lambda_2},,\, X^*_{Y_{\lambda_2},\lambda_1}|\,\lambda_1) &= -11.8821, \\ X^*_{Y_{\lambda_2},\lambda_1} &= (1,\ 2,\ 2,\ 2,\ 2,\ 3,3,\ 3,\ 3,\ 3) \end{aligned}$$

and

$$\begin{aligned} \log P(Y_{\lambda_2},\, X^*_{Y_{\lambda_2},\lambda_2}|\,\lambda_2) &= -7.5601 \\ X^*_{Y_{\lambda_2},\lambda_2} &= (1,\ 2,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3,\ 3) \end{aligned}$$

It is clearly seen, that the Viterbi algorithm gives parallel results with the forward algorithm. The basic difference is only, that in Viterbi algorithm, it is presumed the optimal state path.

3.5 Parameter estimation and Baum-Welch algorithm

The third, and undeniably the most challenging, problem of HMMs is to determine a method to adjust the model parameters $\lambda = (A,\ B,\ \pi(0))$ to satisfy a certain optimization criterion. There exist no known way to analytically solve for the parameter set that maximizes the probability of the observation sequence in a closed form. It is, however, possible to choose the parameter triple $\lambda = (A,\ B,\ \pi(0))$ such that, given an observation sequence Y, the likelihood $P(Y,\ \lambda)$ is locally maximized. Sequence Y is called *the training sequence*. Here, this maximization is performed using iterative procedure such as *the Baum-Welch method*, which is also known as the *maximum likelihood (ML) method*. [7]

It is now presented an auxiliary function, that tries to involve an estimate of the missing information about the state sequence present in HMM.

Given an HMM $\lambda = (A,\ B,\ \pi(0))$ and an observation sequence $Y = (Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N)$, let the set of all possible state sequences be denoted by

$$T = \left\{ X = (X_0 = j_0,\ \ldots\ ,\ X_N = j_N) \in S^{N+1} |\ P(X|\ \lambda) > 0 \right\},$$

and let the size of T be denoted by t_T. Now, the set T can be enumerated as follows

$$T = \left\{ X_s \in S^{N+1} |\ P(X_s|\ \lambda) > 0,\ s = 1,\ 2,\ \ldots\ ,\ t_T \right\}.$$

The joint probability of the observation sequence Y and the possible state sequence $X_s \in T$, on the conditional λ, is denoted by

$$u_s = P(Y,\ X_s|\ \lambda),\ s = 1,\ 2,\ \ldots\ ,\ t_T.$$

For any other model λ^*, it is set

$$v_s = P(Y,\ X_s|\ \lambda^*),\ s = 1,\ 2,\ \ldots\ ,\ t_T.$$

It should be noted, that the probability v_s might equal to zero, since the state paths $X_s \in T$ are concerned with respect to the model λ.

For further considerations, it is done the following *assumption of absolute continuity*: given an observation sequence $Y = (Y_0 = o_0,\ \ldots\ ,\ Y_N = o_N)$, let the corresponding state sequence $X = (X_0 = j_0,\ \ldots\ ,\ X_N = j_N) \in S^{N+1}$ satisfy the condition

$$P(Y,\ X|\ \lambda) = 0.$$

Then, conditioned on the model $\lambda^* = (A^*,\ B^*,\ \pi^*(0))$, the joint probability equals

$$P(Y,\ X|\ \lambda^*) = 0.$$

Lemma 35 *Let $u_s,\ s = 1,\ 2,\ \ldots\ ,\ t_T$ be positive real numbers, and let $u_s,\ s = 1,\ 2,\ \ldots\ ,\ t_T$, be nonnegative real numbers such that $\sum_{s=1}^{t_T} v_s > 0$. Then it follows that*

$$\ln \frac{\sum_{s=1}^{t_T} v_s}{\sum_{s=1}^{t_T} u_s} \geq \frac{\sum_{s=1}^{t_T} (u_s \ln v_s - u_s \ln u_s)}{\sum_{s=1}^{t_T} u_s}.$$

Proof. See [25] and [39].

Theorem 36 *Let the state sequence $X \in T$. Under the assumption of absolute continuity the loglikelihood ratio satisfies the condition*

$$\ln \frac{P(Y,\ X|\ \lambda^*)}{P(Y,\ X|\ \lambda)} \geq \frac{Q(\lambda,\ \lambda^*) - Q(\lambda,\ \lambda)}{P(Y,\ X|\ \lambda)}, \tag{85}$$

where the function Q is defined by

$$Q(\lambda,\ \lambda^*) = Q(\lambda,\ \lambda^*|\ Y) = \sum_{s=1}^{t_T} u_s \ln v_s,$$

and

$$Q(\lambda,\ \lambda) = Q(\lambda,\ \lambda|\ Y) = \sum_{s=1}^{t_T} u_s \ln u_s.$$

Proof. Concerning all possible state sequences in T, the probability of the observation sequence, conditioned on model λ, becomes

$$\begin{aligned} P(Y|\ \lambda) &= \sum_{s=1}^{t_T} P(Y,\ X_s|\ \lambda) \\ &= \sum_{s=1}^{t_T} u_s . \end{aligned}$$

In the same way, under the assumption of absolute continuity and conditioned on the model λ^*, the probability equals

$$\begin{aligned} P(Y|\ \lambda) &= \sum_{s=1}^{t_T} P(Y,\ X_s|\ \lambda^*) \\ &= \sum_{s=1}^{t_T} v_s . \end{aligned}$$

Now, by Lemma 35 we get,

$$\begin{aligned} \ln \frac{P(Y,\ X|\ \lambda^*)}{P(Y,\ X|\ \lambda)} &= \ln \frac{\sum_{s=1}^{t_T} v_s}{\sum_{s=1}^{t_T} u_s} \\ &\geq \frac{\sum_{s=1}^{t_T} (u_s \ln v_s - u_s \ln u_s)}{\sum_{s=1}^{t_T} u_s} \\ &= \frac{Q(\lambda,\ \lambda^*) - Q(\lambda,\ \lambda)}{P(Y,\ X|\ \lambda)} . \end{aligned}$$

If it is possible to find a model λ^* such that the right-hand side of the equation 85 is positive, then by Theorem 36 it is found a better model for the observation sequence Y, in the sense of higher likelihood. The function $Q(\lambda,\ \lambda^*)$ is often called the *quasiloglikelihood* function. By Theorem 36, the largest guaranteed improvement is a result of maximizing the quasiloglikelihood function with respect to λ^*. If for a model λ^* there exists a state sequence X such, that $v_s = 0$, then it is set $Q(\lambda,\ \lambda^*) = -\infty$. This kind

of model will not be a candidate for improvement of likelihood. Model λ is called the *current model* , whereas model λ^* is called the *reestimated model*.

For further developments, it is used following notations:

$$\begin{aligned} n_{ij}(X_s) &= \textit{the number of transitions in the state path } X_s \in T \\ &\quad \textit{from state } i \textit{ to state } j, \\ m_{jk}(X_s) &= \textit{the number of times the symbol } O_k \textit{ is emitted in the} \\ &\quad \textit{state path } X_s \in T. \end{aligned}$$

The indicator function for the initial state j_0 in state sequence $X_s \in T$ is defined by

$$r_j(X_s) = \begin{cases} 1, \ if \ j_0 = j \\ 0, \ if \ j_0 \neq j \end{cases} .$$

Theorem 37 *For an observation sequence* $Y = (Y_0 = o_0, \ ... \ , \ Y_N = o_N)$*, the quasiloglikelihood function equals*

$$Q(\lambda, \ \lambda^*) = \sum_{j=1}^{J} e_j \ln \pi_j^*(0) + \sum_{j=1}^{J}\sum_{k=1}^{K} d_{jk} \ln b_j^*(O_k) + \sum_{j=1}^{J}\sum_{i=1}^{J} c_{ij} \ln a_{ij}^*,$$

where

$$\begin{aligned} e_j &= \sum_{s=1}^{t_T} u_s r_j(X_s), \\ c_{ij} &= \sum_{s=1}^{t_T} u_s n_{ij}(X_s), \end{aligned}$$

and

$$d_{jk} = \sum_{s=1}^{t_T} u_s m_{jk}(X_s).$$

Proof. Let $X_s \in T$ be arbitrary. By Theorem 22, conditioned on the reestimated model λ^*, the joint probability becomes

$$\begin{aligned} v_s &= P(Y, \ X_s | \ \lambda^*) \\ &= \pi_{j_0}^*(0) \prod_{l=0}^{N} b_{j_l}^*(o_l) \prod_{l=1}^{N} a_{j_{l-1} j_l}^*. \end{aligned}$$

Taking logarithm of both sides results

$$\ln v_s = \ln \pi^*_j(0) + \sum_{l=0}^{N} \ln b^*_{j_l}(o_l) + \sum_{l=1}^{N} \ln a^*_{j_{l-1}j_l}.$$

Using this result and regrouping of terms in the summations according to the state transitions and emitted symbols, the quasiloglikelihood function becomes

$$\begin{aligned}
& Q(\lambda,\ \lambda^*) = Q(\lambda,\ \lambda^*|\ Y) = \sum_{s=1}^{t_T} u_s \ln v_s \\
= & \sum_{s=1}^{t_T} u_s \cdot \\
\cdot & \left[\sum_{j=1}^{J} r_j(X_s) \ln \pi^*_j(0) + \sum_{j=1}^{J}\sum_{k=1}^{K} m_{jk}(X_s) \ln b^*_j(O_k) + \sum_{j=1}^{J}\sum_{i=1}^{J} n_{ij}(X_s) \ln a^*_{ij} \right] \\
= & \sum_{j=1}^{J} \left[\sum_{s=1}^{t_T} u_s r_j(X_s) \right] \ln \pi^*_j(0) + \sum_{j=1}^{J}\sum_{k=1}^{K} \left[\sum_{s=1}^{t_T} u_s m_{jk}(X_s) \right] \ln b^*_j(O_k) + \\
& + \sum_{j=1}^{J}\sum_{i=1}^{J} \left[\sum_{s=1}^{t_T} u_s n_{ij}(X_s) \right] \ln a^*_{ij}.
\end{aligned}$$

Now, by using notations

$$\begin{aligned}
e_j &= \sum_{s=1}^{t_T} u_s r_j(X_s), \\
c_{ij} &= \sum_{s=1}^{t_T} u_s n_{ij}(X_s),
\end{aligned}$$

and

$$d_{jk} = \sum_{s=1}^{t_T} u_s m_{jk}(X_s),$$

implies that

$$Q(\lambda,\ \lambda^*) = \sum_{j=1}^{J} e_j \ln \pi^*_j(0) + \sum_{j=1}^{J}\sum_{k=1}^{K} d_{jk} \ln b^*_j(O_k) + \sum_{j=1}^{J}\sum_{i=1}^{J} c_{ij} \ln a^*_{ij}.$$

The quasiloglikelihood function $Q(\lambda,\ \lambda^*)$ may be seen as a sum of finite number expressions separated in the optimization variables. It is now possible to write this optimization problem as follows:

$$\Phi(x) = \sum_{i=1}^{M} c_i \ln x_i = \max!,\ c_i > 0,$$

subject to the constraint

$$\sum_{i=1}^{M} x_i = 1.$$

Lemma 38 *If $c_i > 0,\ i = 1,\ 2,\ \ldots\ ,\ M$, then subject to the constraint*

$$\sum_{i=1}^{M} x_i = 1,$$

the function

$$\Phi(x) = \sum_{i=1}^{M} c_i \ln x_i$$

attains its unique global maximum when

$$x_i = \frac{c_i}{\sum\limits_{i=1}^{M} c_i}.$$

Proof. See [25].

Now, Lemma 38 and Theorem 37 imply the following result.

Corollary 39 *The quasiloglikelihood function $Q(\lambda,\ \lambda^*)$ is maximized, if the reestimated model λ^* is chosen using*

$$\pi_j^*(0) = \frac{e_j}{\sum\limits_{j=1}^{J} e_j}, \tag{86}$$

$$b_j^*(O_k) = \frac{d_{jk}}{\sum\limits_{k=1}^{K} d_{jk}}, \tag{87}$$

and

$$a_{ij}^{*} = \frac{c_{ij}}{\sum_{j=1}^{J} c_{ij}}, \tag{88}$$

where $i,\ j = 1,\ 2,\ \ldots,\ J$ *and* $k = 1,\ 2,\ \ldots,\ K$.

Since each state path $X_s \in T$ begins from some state, then summing over state space S results

$$\sum_{j=1}^{J} r_j(X_s) = 1.$$

By this, it is got

$$\begin{aligned}
\sum_{j=1}^{J} e_j &= \sum_{j=1}^{J}\sum_{s=1}^{t_T} u_s r_j(X_s) \\
&= \sum_{s=1}^{t_T} u_s \sum_{j=1}^{J} r_j(X_s) \\
&= \sum_{s=1}^{t_T} u_s = \sum_{s=1}^{t_T} P(Y,\ X_s|\ \lambda) \\
&= P(Y|\ \lambda).
\end{aligned} \tag{89}$$

Summing over the state space O results

$$\sum_{k=1}^{K} m_{jk}(X_s) = n_j(X_s),$$

which equals the number of times the state path X_s visits state j. Since

$$u_s = P(Y,\ X_s|\ \lambda) = P(X_s|\ Y,\ \lambda)P(Y|\ \lambda),$$

the sum

$$\sum_{k=1}^{K} d_{jk} = \sum_{s=1}^{t_T} u_s \sum_{k=1}^{K} m_{jk}(X_s) = \sum_{s=1}^{t_T} u_s n_j(X_s) = n_j \tag{90}$$

is proportional to the expected number of visits to state j, given the emitted sequence Y and the model λ. By the help of equations 89 and 90, the estimation formulas 86 - 88 can be written

$$\pi_j^{*}(0) = \frac{e_j}{P(Y|\ \lambda)}, \tag{91}$$

$$b_j^{*}(O_k) = \frac{d_{jk}}{n_j}, \tag{92}$$

and

$$a_{ij}^{*} = \frac{c_{ij}}{\sum_{j=1}^{J} c_{ij}}, \tag{93}$$

where $i,\ j\ =\ 1,\ 2,\ \ldots\ ,\ J$ and $k\ =\ 1,\ 2,\ \ldots\ ,\ K$. It is easily seen, that equations 91 - 93 satisfy the natural stochastic constraints, which are demanded of an HMM. This confirms, that equations 91 - 93 give a new well-defined model $\lambda^{*} = (A^{*},\ B^{*},\ \pi^{*}(0))$.

3.6 Reestimates in forward and backward variables

The Baum-Welch algorithm is essentially exploiting the forward and backward variables for the estimation problem. To show that this is possible, it is first presented the reestimation formulas 91 - 93 in smoothing probabilities

$$\hat{\pi}_{j}\ (n|\ N) = P(X_n = j|\ Y_0 = o_0,\ \ldots,\ Y_N = o_N), \tag{94}$$

where $j\ =\ 1,\ 2,\ \ldots\ ,\ J$ and $0\ \leq\ n\ \leq\ N$. In addition, it is used a new *smoothing transition probability*

$$\hat{\pi}_{ij}\ (n|\ N) = P(X_n = j,\ X_{n+1} = j|\ Y_0 = o_0,\ \ldots,\ Y_N = o_N), \tag{95}$$

where $j\ =\ 1,\ 2,\ \ldots\ ,\ J$ and $0\ \leq\ n\ \leq\ N\ -\ 1$. By the Definition 95, it is immediately seen the result of the following lemma.

Lemma 40 *The smoothing probability equals*

$$\hat{\pi}_{j}\ (n|\ N) = \sum_{j=1}^{J} \hat{\pi}_{ij}\ (n|\ N), \tag{96}$$

for all $j\ =\ 1,\ 2,\ \ldots\ ,\ J$ *and* $0\ \leq\ n\ \leq\ N\ -\ 1$.

Theorem 41 *Under the current model* λ*, the initial state probability equals*

$$\pi_{i}^{*} = \hat{\pi}_{i}\ (0|\ N),$$

for all $i\ =\ 1,\ 2,\ \ldots\ ,\ J$.

Proof. Let $i = 1, 2, \ldots, J$ and $n = 0, 1, \ldots, N-1$ be arbitrary. It is first defined an indicator function

$$I_{ij}(n) = \begin{cases} 1, \ if \ X_n = i, \ X_{n+1} = j \\ 0, \ otherwise \end{cases}$$

$i, j = 1, 2, \ldots, J, \ 0 \leq n \leq N-1$. By the definition of conditional expectation of a discrete random variable the conditional expectation equals

$$\begin{aligned} E\left[I_{ij}(n)|\, Y\right] &= P(X_n = j, \, X_{n+1} = j|\, Y) \\ &= \overset{\wedge}{\pi}_{ij}\,(n|\, N). \end{aligned}$$

Thus, by Lemma 40 and linearity of conditional expectation, the smoothing probability becomes

$$\begin{aligned} \overset{\wedge}{\pi}_i\,(n|\, N) &= \sum_{j=1}^{J} E\left[I_{ij}(n)|\, Y\right] \\ &= E\left[Z_i(n)|\, Y\right], \end{aligned}$$

where

$$Z_i(n) = \sum_{j=1}^{J} I_{ij}(n).$$

The generic outcome of the random variable $Z_i(n)$ is the number of times a state path visits state i at time instant n. It is thus clear, that the smoothing probability $\overset{\wedge}{\pi}_i\,(n|\, N)$ may be interpreted as the expected number of times the state i is visited at time n by the state paths of the underlying Markov chain, given the observation sequence Y and model λ. The random variable $Z_i(n)$ may be described using the set of possible states $T = \left\{X_s \in S^{N+1}|\ P(X_s|\, \lambda) > 0, \ s = 1, \ 2, \ \ldots, \ t_T\right\}$, by defining for each state sequence X_s a new random variable

$$z_i(n, s) = \sum_{j=1}^{J} I_{ij}(n).$$

By this the random variable equals

$$Z_i(n) = \sum_{s=1}^{t_T} z_i(n, s).$$

The linearity of expectation, definitions of expectation and conditional probability imply

$$\begin{aligned} E\left[Z_i(n)|\,Y\right] &= E\left[\sum_{s=1}^{t_T} z_i(n,s)|\,Y\right] = \sum_{s=1}^{t_T} E\left[z_i(n,s)|\,Y\right] \\ &= \sum_{s=1}^{t_T} z_i(n,s)\,P(X_s|\,Y) = \sum_{s=1}^{t_T} z_i(n,s)\,\frac{P(X_s,\,Y)}{P(Y)} \\ &= \frac{1}{P(Y)}\sum_{s=1}^{t_T} z_i(n,s)\,u_s = C\sum_{s=1}^{t_T} z_i(n,s)\,u_s. \end{aligned} \tag{97}$$

Setting $n = 0$, gives

$$\begin{aligned} \overset{\wedge}{\pi}_i\,(0|\,N) &= \frac{\sum_{s=1}^{t_T} z_i(0,s)\,u_s}{P(Y)} = \frac{\sum_{s=1}^{t_T} r_i(X_s)\,u_s}{P(Y)} \\ &= \frac{e_i}{P(Y)} = \pi_i^*, \end{aligned}$$

for all $i = 1,\,2,\,\ldots,\,J$ and $n = 0,\,1,\,\ldots,\,N-1$.

It was stated below, that the smoothing probability $\overset{\wedge}{\pi}_i\,(n|\,N)$ may be interpreted as the expected number of times the state i is visited at time n among the allowed state paths of the hidden Markov chain, given the observation sequence Y and model λ. Thus the reestimate $\pi_i^* = \overset{\wedge}{\pi}_i\,(0|\,N)$ may be interpreted as the expected frequency of starting at state i given the observation sequence Y and model λ.

Theorem 42 *Under the current model λ, the conditional probability equals*

$$b_j^*(O_k) = \frac{\sum_{n=0}^{N} I(n,\,O_k)\;\overset{\wedge}{\pi}_j\,(n|\,N)}{\sum_{n=0}^{N} \overset{\wedge}{\pi}_j\,(n|\,N)},$$

where $I(n,\,O_k) = \begin{cases} 1, & if\ Y_n = O_k \\ 0, & otherwise \end{cases}$ *, for all* $n = 0,\,1,\,\ldots,\,N$, *and* $k = 1,\,2,\,\ldots,\,K$.

Proof. Let $j = 1, 2, \ldots, J$ and $k = 1, 2, \ldots, K$ be arbitrary. Consider the following indicator function

$$I_j(n, O_k) = \begin{cases} 1, \; if \; X_n = j, \; Y_n = O_k \\ 0, \; otherwise \end{cases} .$$

This can be written as the product

$$I_j(n, O_k) = I_j(n)\, I(n, O_k), \tag{98}$$

where

$$I_j(n) = \begin{cases} 1, \; if \; X_n = j \\ 0, \; otherwise \end{cases} .$$

Using equation 98 and the fact, that each observation O_k is included in the observation sequence Y, the sum equals

$$\begin{aligned} \sum_{n=0}^{N} P(X_n &= j, Y_n = O_k | \, Y) = \sum_{n=0}^{N} E\left[I_j(n, O_k) | \, Y\right] \\ &= \sum_{n=0}^{N} E\left[I_j(n)\, I(n, O_k) | \, Y\right] \\ &= \sum_{n=0}^{N} I(n, O_k)\, E\left[I_j(n) \, | \, Y\right] \\ &= \sum_{n=0}^{N} I(n, O_k)\, P(X_n = j | \, Y) \\ &= \sum_{n=0}^{N} I(n, O_k)\, \hat{\pi}_i \, (n | \, N) \end{aligned} \tag{99}$$

Again, the random variable $\sum_{n=0}^{N} I_j(n, O_k)$ may be described using the set of possible state sequences

$$T = \left\{ X_s \in S^{N+1} | \; P(X_s | \; \lambda) > 0, \; s = 1, \; 2, \; \ldots, \; t_T \right\},$$

by defining for each state sequence X_s a new random variable

$$z_j(s, O_k) = \sum_{n=0}^{N} I_j(n, O_k).$$

Considering the whole set T, the indicator function becomes

$$\sum_{n=0}^{N} I_j(n,\, O_k) = \sum_{s=1}^{t_T} z_j(s, O_k).$$

Thus

$$\begin{aligned}
&\sum_{n=0}^{N} P(X_n = j,\, Y_n = O_k |\, Y) = \sum_{n=0}^{N} E\left[I_j(n,\, O_k) |\, Y\right] \\
= \;& E\left[\sum_{n=0}^{N} I_j(n,\, O_k) |\, Y\right] = E\left[\sum_{s=1}^{t_T} z_j(s, O_k) |\, Y\right] \\
= \;& \sum_{s=1}^{t_T} E\left[z_j(s, O_k) |\, Y\right] = \sum_{s=1}^{t_T} z_j(s, O_k)\, P(X_s |\, Y) \\
= \;& \sum_{s=1}^{t_T} z_j(s, O_k)\, \frac{P(X_s, Y)}{P(Y)} = \frac{1}{P(Y)} \sum_{s=1}^{t_T} z_j(s, O_k)\, P(X_s, Y) \\
= \;& C \sum_{s=1}^{t_T} m_{jk}(X_s)\, u_s = C\, d_{jk}. \qquad (100)
\end{aligned}$$

Now by summing over times $n = 0,\, 1,\, \ldots,\, N$, and equations 90 and 97 result the following

$$\begin{aligned}
\sum_{n=0}^{N} \overset{\wedge}{\pi}_j\, (n |\, N) &= \sum_{n=0}^{N} E\left[Z_j(n) |\, Y\right] = \sum_{n=0}^{N} C \sum_{s=1}^{t_T} z_j(n, s)\, u_s \\
&= C \sum_{s=1}^{t_T} \sum_{n=0}^{N} z_j(n, s)\, u_s = C \sum_{s=1}^{t_T} n_j(s)\, u_s \\
&= C\, n_j. \qquad (101)
\end{aligned}$$

Using equations 100 and 101 gives the required result.

The practical interpretation of the reestimate probability $b_j^*(O_k)$ can be done by the property of the smoothing probability $\overset{\wedge}{\pi}_j\, (n |\, N)$. Summing over times $n = 0,\, 1,\, \ldots,\, N$, the reestimate $b_j^*(O_k)$ may be seen as a ratio of the expected number of visits in state j and emitting the symbol O_k, and the expected number of transitions from state j, both expectations conditioned on the observation sequence Y and model λ.

Theorem 43 *Under the current model λ, the transition probability equals*

$$a_{ij}^{*} = \frac{\sum_{n=0}^{N-1} \overset{\wedge}{\pi}_{ij}\,(n|\,N)}{\sum_{n=0}^{N-1} \overset{\wedge}{\pi}_{i}\,(n|\,N)},$$

for all $i,\ j = 1,\ 2,\ \ldots,\ J$.

Proof. Let $i,\ j = 1,\ 2,\ \ldots,\ J$ be arbitrary. It is first defined an indicator function

$$I(n,\ i,\ j) = \begin{cases} 1,\ if\ X_n = i,\ X_{n+1} = j \\ 0,\ otherwise \end{cases},\ n = 0,\ 1,\ \ldots, N-1.$$

Definitions of conditional expectation and smoothing transition probability give

$$\begin{aligned} E\left[I(n,i,\ j)|\,Y\right] &= P(X_n = i,\ X_{n+1} = j|\,Y) \\ &= \overset{\wedge}{\pi}_{ij}\,(n|\,N). \end{aligned}$$

By the linearity of expectation, the sum equals

$$\sum_{n=0}^{N-1} \overset{\wedge}{\pi}_{ij}\,(n|\,N) = \sum_{n=0}^{N-1} E\left[I(n,i,\ j)|\,Y\right] = E\left[\sum_{n=0}^{N-1} I(n,i,\ j)|\,Y\right].$$

For each state sequence X_s, it is defined a random variable

$$z(n,\ s,\ i,\ j) = I(n,i,\ j).$$

This, linearity of expectation and definition of conditional probability imply

$$\begin{aligned} \sum_{n=0}^{N-1} \overset{\wedge}{\pi}_{ij}\,(n|\,N) &= \sum_{n=0}^{N-1} E\left[I(n,i,\ j)|\,Y\right] = \sum_{n=0}^{N-1} E\left[\sum_{s=1}^{t_T} z(n,s,\ i,\ j)|\,Y\right] \\ &= \sum_{n=0}^{N-1}\sum_{s=1}^{t_T} E\left[z(n,s,\ i,\ j)|\,Y\right] \\ &= \sum_{n=0}^{N-1}\sum_{s=1}^{t_T} z(n,s,\ i,\ j)\,P(X_s|\,Y) \\ &= \frac{1}{P(Y)}\sum_{s=1}^{t_T} P(X_s,\ Y)\sum_{n=0}^{N-1} z(n,s,\ i,\ j) \\ &= C\sum_{s=1}^{t_T} u_s\,n_{ij}(s) = C\,c_{ij}. \end{aligned} \tag{102}$$

By, Lemma 40 the sum equals

$$\begin{aligned}\sum_{n=0}^{N-1} \overset{\wedge}{\pi}_i\,(n|\,N) &= \sum_{n=0}^{N-1}\sum_{j=1}^{J} \overset{\wedge}{\pi}_{ij}\,(n|\,N) \\ &= \sum_{j=1}^{J}\sum_{n=0}^{N-1} \overset{\wedge}{\pi}_{ij}\,(n|\,N) \\ &= \sum_{j=1}^{J} C\,c_{ij} = C\sum_{j=1}^{J} c_{ij}. \qquad (103)\end{aligned}$$

Now, equations 102, 103 give the required result.

It was seen in Theorem 41, that the smoothing transition probability $\overset{\wedge}{\pi}_{ij}\,(n|\,N)$ equals the expected number of making transitions from state i to state j at time n, conditioned on the observation sequence Y and model λ. Additionally, the characterization of the smoothing probability $\overset{\wedge}{\pi}_i\,(n|\,N)$ together with summing over times $n = 0,\,1,\,\ldots\,N-1$, gives the interpretation of the reestimate probability a^*_{ij} as the ratio of the expected number of transitions from state i to state j and the expected number of transitions from state i, both expectations conditioned on the observation sequence Y and model λ.

To present the reestimation formulas in computationally effective form with forward and backward variables, the smoothing probabilities have to be written in terms of the forward and backward variables.

Lemma 44 *Under the current model λ, for the given observation sequence Y the sum of smoothing transition probabilities equals*

$$\sum_{n=0}^{N-1} \overset{\wedge}{\pi}_{ij}\,(n|\,N) = \frac{\sum_{n=0}^{N-1} \alpha_n(i)\,a_{ij}\,b_j(o_{n+1})\,\beta_{n+1}(j)}{P(Y)}.$$

Proof. Let $i,\,j = 1,\,2,\,\ldots\,,\,J$ be arbitrary. By the definitions of smoothing and conditional probabilities, the sum of smoothing transition probabilities equals

$$\sum_{n=0}^{N-1} \overset{\wedge}{\pi}_{ij}\,(n|\,N) = \sum_{n=0}^{N-1} P(X_n = i,\,X_{n+1} = j|\,Y)$$

$$
\begin{aligned}
&= \sum_{n=0}^{N-1} \frac{P(X_n = i,\, X_{n+1} = j,\, Y)}{P(Y)} \\
&= \frac{1}{P(Y)} \sum_{n=0}^{N-1} P(Y \,|\, X_n = i,\, X_{n+1} = j)\, P(X_n = i,\, X_{n+1} = j) \\
&= C \sum_{n=0}^{N-1} P(Y \,|\, X_n = i,\, X_{n+1} = j)\, P(\, X_{n+1} = j \,|\, X_n = i)\, P(\, X_n = i) \\
&= C \sum_{n=0}^{N-1} P(Y \,|\, X_n = i,\, X_{n+1} = j)\, a_{ij}\, P(\, X_n = i). \qquad (104)
\end{aligned}
$$

But Theorems 23, 27 and definitions of forward and backward variables imply, that

$$
\begin{aligned}
& P(Y \,|\, X_n = i,\, X_{n+1} = j) \\
= \;& P(Y_0 = o_0,\, \ldots,\, Y_n = o_n \,|\, X_n = i) \cdot \\
& \cdot P(Y_{n+1} = o_{n+1},\, \ldots,\, Y_N = o_N \,|\, X_{n+1} = j) \\
= \;& P(Y_0 = o_0,\, \ldots,\, Y_n = o_n \,|\, X_n = i)\, P(Y_{n+1} = o_{n+1} \,|\, X_{n+1} = j) \cdot \\
& \cdot P(Y_{n+2} = o_{n+2},\, \ldots,\, Y_N = o_N \,|\, X_{n+1} = j) \\
= \;& \alpha_n(i)\; b_j(o_{n+1})\, \beta_{n+1}(j). \qquad (105)
\end{aligned}
$$

Thus, using equations 104 and 105 imply, that

$$
\sum_{n=0}^{N-1} \overset{\wedge}{\pi}_{ij}\, (n \,|\, N) = \frac{\sum_{n=0}^{N-1} \alpha_n(i)\, a_{ij}\, b_j(o_{n+1})\, \beta_{n+1}(j)}{P(Y)}.
$$

Finally, Theorems 41 - 43 and Lemma 44 imply the following corollary.

Corollary 45 (Baum-Welch Reestimation Formulas) *Under the model $\lambda = (A,\, B,\, \pi(0))$, and given an observation sequence Y, a new reestimate model is defined by the following equations:*

1. For $j = 1,\, 2,\, \ldots\, ,\, J$, the initial state probability equals

$$
\pi_j^* = \frac{\alpha_0(i)\; \beta_0(j)}{P(Y)}. \qquad (106)
$$

2. For $i,\ j\ =\ 1,\ 2,\ \ldots\ ,\ J$, and $k\ =\ 1,\ 2,\ \ldots\ ,\ K$, the emission probability equals

$$b_j^*(O_k) = \frac{\sum\limits_{n=0}^{N} I(n,\ O_k)\,\alpha_n(j)\,\beta_n(j)}{\sum\limits_{n=0}^{N} \alpha_n(j)\ \beta_n(j)}. \tag{107}$$

3. For $i, j = 1,\ 2,\ \ldots\ ,\ J$, the transition probability equals

$$a_{ij}^* = \frac{\sum\limits_{n=0}^{N-1} \alpha_n(i)\ a_{ij}\, b_j(o_{n+1})\beta_{n+1}(j)}{\sum\limits_{n=0}^{N-1} \alpha_n(i)\ \beta_n(i)}. \tag{108}$$

It is now shown, that the Baum-Welch reestimation formulas 106 - 108 maximize the quasiloglikelihood $Q(\lambda,\ \lambda^*)$, given a current model λ and an observation sequence Y. The Baum-Welch algorithm is clearly an iterative method. If the current model λ is set to $\lambda^{(k)}$, and the reestimate model λ^* is set to $\lambda^{(k+1)}$, then the equations 106 - 108 may be written in one formula as

$$\lambda^{(k+1)} = \tau(\lambda^{(k)}),\ k = 0,\ 1,\ 2,\ \ldots \tag{109}$$

It is said that the B-W algorithm has converged, if

$$\lambda^* = \tau(\lambda^*).$$

Then model λ^* is called a *fixed point of* τ. In the analysis of this iteration process, it is to find how the fixed points of the algorithm $\lambda^* = \tau(\lambda^*)$ are related to finding the maximum likelihood

$$\overset{\wedge}{\lambda}_{ML} = \arg\max_{\lambda} P(Y|\,\lambda) = \arg\max_{\lambda} L(\lambda).$$

Theorem 46 *For every HMM λ, it holds*

$$L(\tau(\lambda)) \geq L(\lambda),$$

with equality , if and only if λ is a critical point of $L(\lambda)$, or equivalently is a fixed point of τ.

Proof. See [7].

Previous theorem says, that each B-W iteration improves the model strictly in terms of the likelihood, unless a critical point, or equivalently a fixed point, is met. It should be pointed out, that this confirms the iteration ends to a local maximum, only. The final result of this reestimation procedure is an *ML* estimate of the HMM. In practical situation, it is often needed the knowledge of strict improvement, only. Then, the estimation process is cut after some predefined amount of iterations.

3.7 Implementation issues for HMMs

3.7.1 Scaling

From the foregoing presentations, one may easily think, that the problems of hidden Markov modeling can be solved by straightforward translation of the formulas into computer programs. Any of the methods applied for the classification or the estimation problem, require evaluation of the forward variable $\alpha_n(i)$ and the backward variable $\beta_n(i),\ n = 1, 2, \ldots, N,\ i = 1, 2, \ldots, J$. From the recursive formulas for these quantities, 77 and 80, it can be seen, that as $N \to \infty$, both $\alpha_N(i) \to 0$, and $\beta_1(i) \to 0$ in an exponential fashion. For enough large N, this fact will lead to the underflow on any real computer, if equations 77 and 80 are evaluated directly. [25]

There has been found a simple method for scaling these computations so that the underflow is avoided. The principle of this scaling procedure is based on multiplying each forward variable $\alpha_n(i)$ and backward variable $\beta_n(i)$ by scaling coefficient, that is independent of state i. At the end of computation, the total effect of the scaling is removed. The discussions in [38] show, that a reasonable scaling procedure is to compute both $\alpha_n(i)$ and $\beta_n(i)$ according to formulas 77 and 80, and then multiply both of them with the same scaling coefficient

$$c_n = \frac{1}{\sum_{i=1}^{J} \alpha_n(i)}.$$

At each time instant $n = 0,\ 1,\ \ldots\ ,\ N$, this results the scaled variables

$$\begin{cases} \hat{\alpha}_n\ (i) = c_n\, \alpha_n(i) \\ \hat{\beta}_n\ (i) = c_n\, \beta_n(i) \end{cases},\ i = 1,\ 2,\ \ldots\ ,\ J,$$

which are then used as the forward and backward variables in formulas 77 and 80 at the next time instant $n+1$. It is shown in [38], that the total effect of this kind of scaling is completely cancelled, when these scaled variables are applied to the reestimation formulas 106 - 108.

3.7.2 Multiple observation sequences

It the following, it is concentrated on the left-right HMMs, that are the main interest of the experimental part of this work. In a left-right HMM, the hidden states proceed from state 1 at time instant $n = 0$, to state J, at time instant $n = N$, in a sequential manner. This imposes constraints on the transition matrix A, and the initial state distribution $\pi(0)$. The transient nature of the hidden states allows only a small number of observations for any state until to the transition to a successor state. One way to solve this problem, is to use multiple observation sequences.

Let the set of R observation sequences be denoted as

$$Y = \left\{ Y^1,\ Y^2,\ \ldots\ ,\ Y^R \right\},$$

where

$$Y^r = \left\{ Y_1^r = o_1^r,\ Y_2^r = o_2^r,\ \ldots\ ,\ Y_{N_r}^r = o_{N_r}^r \right\},$$

and

$$o_j^r \in \{O_1,\ O_2, \ldots\ ,\ O_K\},$$

$r = 1,\ 2,\ \ldots\ ,\ R,\ j = 1,\ 2, \ldots\ ,\ N_r$. It is assumed, that each observation sequence is independent of every other observation sequence. The goal of the estimation, is to find the model λ, to maximize the probability

$$P(Y \mid \lambda) = \prod_{r=1}^{R} P(Y^r \mid \lambda) = \prod_{r=1}^{R} P_r.$$

For a left-right HMM the initial state distribution is fixed by the fact, that the underlying hidden Markov chain is always forced to start at state 1. This

makes the reestimation of the initial state distribution unnecessary. Each numerator and denominator of the reestimation formulas 107 - 108 represents an average number of some event related to to the given observation sequence Y and current model λ. Accordingly, it makes sense simply to sum these events over all observation sequences. By this consideration, the modified reestimation formulas are of form:

For $i,\ j\ =\ 1,\ 2,\ \ldots\ ,\ J$, and $k\ =\ 1,\ 2,\ \ldots\ ,\ K$, the emission probability equals

$$b_j^*(O_k) = \frac{\sum\limits_{r=1}^{R} \frac{1}{P_r} \sum\limits_{n=0}^{N} I(n,\ O_k)\, \alpha_n(j)\ \beta_n(j)}{\sum\limits_{r=1}^{R} \frac{1}{P_r} \sum\limits_{n=0}^{N} \alpha_n(j)\ \beta_n(j)}. \tag{110}$$

For $i, j\ =\ 1,\ 2,\ \ldots\ ,\ J$, the transition probability equals

$$a_{ij}^* = \frac{\sum\limits_{r=1}^{R} \frac{1}{P_r} \sum\limits_{n=0}^{N-1} \alpha_n(i)\ a_{ij}\, b_j(o_{n+1})\beta_{n+1}(j)}{\sum\limits_{r=1}^{R} \frac{1}{P_r} \sum\limits_{n=0}^{N-1} \alpha_n(i)\ \beta_n(i)}. \tag{111}$$

Again, the use of scaled forward and backward variables are cancelled exactly. [38, pp. 369 - 370]

4 Bayes classification and context-dependency

The statistical nature of the generated features in a PR system are due to the statistical variation of patterns as well as noise in the measuring sensors. Adopting this, gives a reason to design classifiers that classify an unknown pattern in the most probable of the possible classes. Given a classification task of M classes $C_1,\ C_2,\ \ldots\ ,\ C_M$, and an unknown pattern, presented by a feature vector x, it is formed the M conditional probabilities $P(C_i|\, x),\ i = 1,\ 2,\ \ldots\ ,\ M$, which are referred to as *a posteriori probabilities*. Each of these probabilities represents the probability, that the unknown pattern, presented by a feature vector x, belongs to the respective class C_i. The problem of classification is now to find the maximum probability for the most probable class or, equivalently the maximum of an appropriately defined function of the conditional probabilities. The unknown pattern is then classified to the class corresponding to this maximum.

Let us focus on the two-class case. It is assumed, that the patterns to be classified, belong to two classes, denoted by C_1 and C_2. For the future, it is made an assumption, that the *a priori probabilities* $P(C_i),\ i = 1,\ 2$ are known. This assumption makes sense, because even if these probabilities are not known, it is possible to estimate them from the available training vectors. Also, the class-conditional probability density functions $p(x|\, C_i),\ i = 1,\ 2$ are are assumed to be known. These functions describe the *distribution of feature vectors* in each of the two classes. If they are not known, they can be estimated from the available training data, also. The probability density function $p(x|\, C_i)$ is sometimes called the *likelihood function of the class C_i with respect to pattern x*. In this work, it is considered the feature vectors, that get only discrete values. In this case, the density function $p(x|\, C_i)$ is denoted by $P(x|\, C_i)$. Now, using the Bayes rule presented in equation 3 makes it possible to compute the conditional probabilities $P(C_i|\, x)$ as follows

$$\begin{aligned} P(C_i|\, x) &= \frac{P(C_i)\, P(x|\, C_i)}{P(x)} \\ &= \frac{P(C_i)\, P(x|\, C_i)}{\sum_{k=1}^{2} P(C_k) P(x|\, C_k)}. \end{aligned}$$

Thus, the *Bayes classification rule* can be stated as:

- if $P(C_1|\, x) > P(C_2|\, x)$, x is classified to class C_1,

- if $P(C_2|\, x) > P(C_1|\, x)$, x is classified to class C_2.

In the case of equality, the pattern may be assigned to either of the classes. The probability density function $P(x)$ is same for all classes and may be cancelled in the decision process. Furthermore, if the *a priori* probabilities $P(C_i)$ are equal, the Bayes classification rule can formulated as

- if $P(x|\, C_1) > P(x|\, C_2)$, x is classified to class C_1,

- if $P(x|\, C_2) > P(x|\, C_1)$, x is classified to class C_2

It may happen that the probabilistic quantities used to the classification are numerically small. To handle this problem, it is more practical to use the logarithms of these probabilities.

In a classification task with M classes, C_1, C_2, $\ldots$, C_M, an unknown pattern, presented by a feature vector x, is assigned to class C_i, if

$$P(x|\, C_i) > P(x|\, C_j),$$

for all $j \neq i$, or equivalently, if

$$\log P(x|\, C_i) > \log P(x|\, C_j),$$

for all $j \neq i$.

So far, it has been introduced the principle of Bayes classification in a case where each pattern is characterized as one feature vector, and no relations between the various classes exist. This kind of situation is called the *context-free classification*. The idea of hidden Markov models is to deal with whole sequences of information, in other words, sequences of feature vectors. At the same time, it is done an assumption, that successive feature vectors are not independent. This is clear by the Markov property of the underlying hidden Markov chain. Under such an assumption, classifying each feature vector separately from the others has obviously no sense. Now, the classification of a single feature vector depends on its own

value, on the values of the other feature vectors, and on the existing relations among the various classes. By this description it is reasonable to call this kind of classification as *context-dependent classification*. Typical situations of context-dependency appear in applications such as image processing, communications, and speech recognition. The *mutual information* residing within successive feature vectors requires the classification task to be performed using all vectors simultaneously and to be arranged in the same sequence in which they were observed in the experiments. This gives a reason to refer to the feature vectors as successive *observations* producing an *observation sequence*.

In the following it is assumed, that each reference HMM will be treated as a distinct class. Let those classes be $\lambda_1, \lambda_2, \ldots, \lambda_M$. The sequence of observations Y is a result of emissions, due to the transitions among the different states of the respective models. Given a sequence of $N+1$ observations $Y = (Y_0 = o_0, \ \ldots \ , \ Y_N = o_N)$, resulting from an unknown pattern, it is to decide to which class the sequence belongs. The Bayesian classifier finds the class λ^*, for which

$$\lambda^* = \arg \max_{\lambda_i} P(\lambda_i | \, Y).$$

For equiprobable reference models or classes this is equivalent to

$$\lambda^* = \arg \max_{\lambda_i} P(Y | \, \lambda_i),$$

or

$$\lambda^* = \arg \max_{\lambda_i} \left(\log P(Y | \, \lambda_i) \right).$$

The computation of the required probabilities for the classification or *recognition* is straightforwardly solved by forward and backward algorithms. An alternative and very often used approach to perform the classification is to use the Viterbi algorithm. According to this method, for a given observation sequence Y the most probable state sequence X_i^* is computed against each reference model λ_i for which

$$X_i^* = \arg \max_{all \, X} P(Y, \, X | \, \lambda_i).$$

In the sense of Bayes classification the unknown pattern is then classified to model λ^*, for which

$$\lambda^* = \arg \max_{\lambda_i} P(Y, \, X_i^* | \, \lambda_i),$$

or by the log-Viterbi algorithm

$$\lambda^* = \arg \max_{\lambda_i} \log P(Y, X_i^* | \lambda_i).$$

This principle of classification is applied in the HMM recognition system of this work. [44, pp. 13 - 17, 307 - 321]

Classical way of presenting classification results in an M class classification task, is to use *a confusion table*. A confusion table may be seen as a matrix

$$(x_{ij})_{i=1,\,j=1}^{M,\,M},$$

where an element equals

$$\begin{aligned} x_{ij} \quad = \quad & the\ number\ of\ objects,\ that\ are \\ & known\ to\ belong\ to\ class\ i,\ and\ are \\ & classified\ to\ class\ j. \end{aligned}$$

Diagonal elements of the confusion table give the numbers of correct classifications, whereas nondiagonal elements give the numbers of misclassifications.

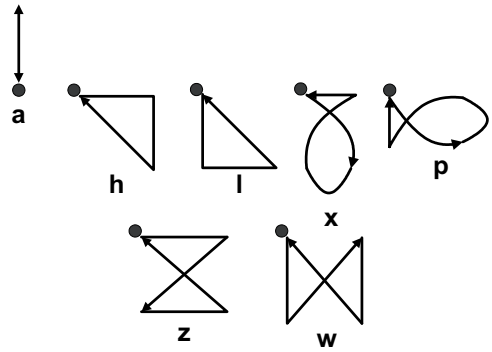

Figure 6. An illustration of the modified unistroke alphabets. (Center of action marked with a spot)

5 HMM recognition system

5.1 Measurements

Second purpose of this work is to show that discrete HMMs, can be used to construct a reliable offline hand gesture recognition system. For one person, it was given a tilted board with a fixed and marked center of action. Holding a penlike device in his right hand the person makes predefined gestures around the marked point, always beginning and ending at this same point. Between separate repeatings of the same gesture, the movement of the hand is stopped for a while. The gestures are modified from the unistroke alphabets which are used in PDAs for text entry. The set of gestures is illustrated in Figure 6. Every gesture was repeated 140 times, consisting of 90 training examples and 50 test examples. [16, pp. 59 - 60]

Acceleration of the penlike device in three orthogonal directions is measured with a 3-dimensional accelerometer system. The system consists of one 2-D and one 1-D accelerometer, which are attached to the handheld device. The accelerometers are manufactured by Analog Devices and their type is ADXL202. The analog signal from the accelerometers is A/D-converted and sampled with 80 Hz frequency, with National Instruments DaqCard 1200 measurement board which is connected to a PC-CARD slot in a laptop PC. The measurement program that stores the digitized acceleration signals in the disk is programmed in LabView which is a graphical

programming environment from National Instruments. The quality of the signals can be checked visually with the measurement program before starting the recording.

5.2 Preprocessing

After sampling and digitation, collected acceleration signals are moved to MATLAB (version 5.3) environment for further processing. Each component of the three dimensional acceleration signals are filtered with fourth-order lowpass Butterworth filter with the 3 dB cut-off frequency of 4 Hz. The filtered signals are segmented automatically into separate 3-dimensional acceleration segments corresponding to individual examples of the gestures. The automation is simply based on the fact, that the hand movement was stopped between separate examples of gestures. One example of a gesture is referred to as a *sample*. This process of segmentation gives the reason to call the recognition system as an *isolated* recognition system.

5.3 Feature extraction

The effects of feature extraction was explained in section 1.4. In this work, it is used two ways of generating features from 3-dimensional acceleration signals. Each sample may be seen as a sequence of 3-dimensional acceleration vectors. This sequence is converted to a corresponding sequence of feature vectors in the following two ways:

Feature 1. The three dimensional acceleration signal is resampled taking every fifth sample point. This results a 3-dimensional feature vector representing the signal level fluctuation during a gesture.

Feature 2. A sample is devided into successive windows with overlap of 50 %. This is illustrated in Figure 7. In a window for each component of acceleration it is calculated three parameters:

Parameter 1: sample mean,

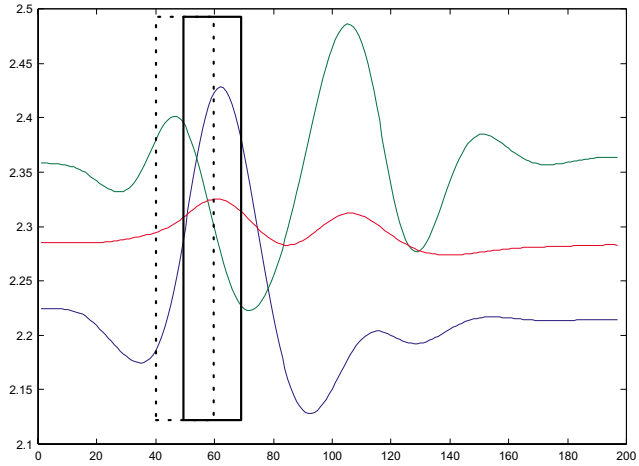

Figure 7. Two successive windows in one sample corresponding to alphabet p.

Parameter 2: difference between maximum and minimum signal values and,

Parameter 3: sample standard deviation.

This process results a 9-dimensional feature vector.

After feature extraction, it is performed the *normalisation of all feature vectors* to have 0 *sample mean* and 1 *sample variance* with respect to each component. The normalization parameters are calculated using the training set. With sample standard deviation it is meant the square root of the sample variance. For the definitions of sample mean, and sample variance, it is referred to [21, p. 1216]. The whole process for data preprocessing and feature extraction is illustrated in Figure 8.

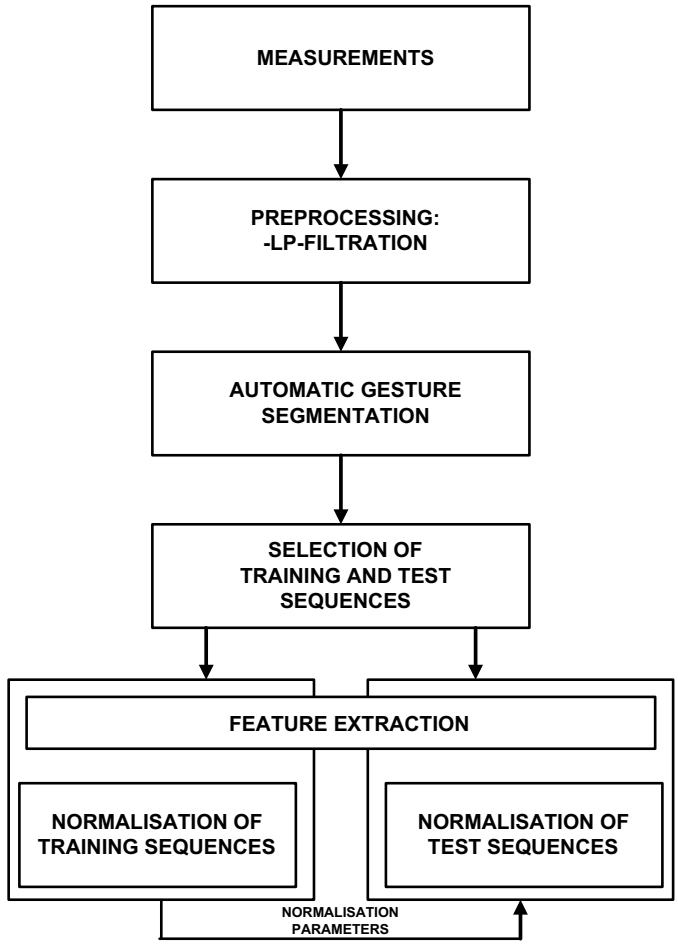

Figure 8. Block diagram of preprocessing and feature extraction of acceleration signals in MATLAB environment.

5.4 Vector quantization

Unsupervised learning or clustering techniques are widely used methods in pattern recognition and neural networks for exploratory data analysis. These methods are often used to understand the spatial structure of the data samples and/or to reduce the computational costs of designing a classifier. A common goal of unsupervised learning algorithms is to distribute a certain number of reference vectors in a possibly high dimensional space according to some quality criteria. This is often called *vector quantization* (VQ). The typical VQ-problem is defined as follows: Given a finite data set $S = \{x_1, x_2, \ldots, x_N\}$, $x_i \in R^d$, where x_i are independently and identically distributed according to some probability distribution $p(x)$, find a set of reference vectors $A = \{c_1, c_2, \ldots, c_m\}$, $c_i \in R^d$, such that a given distortion measure $E(p(x), A)$ is minimized. A typical application is, for example, to compress a set of vectors for transmission purpose with vector quantization. This can be achieved with VQ, which minimizes the expected quantization error

$$E(p(x), A) = \sum_{i=1}^{m} \int_{S_i} \|x - c_i\|^2 p(x) dx, \tag{112}$$

by positioning the c_i, where $S_i = \left\{ x \in R^d | \, i = \arg\min_{j \in \{1,2,\ldots,m\}} \|x - c_i\| \right\}$ is the Voronoi region of a vector c_i and $\|\cdot\|$ denotes the Euclidean distance. In practice, the only knowledge about the distribution $p(x)$ is the data set S. Thus, it is possible to minimize only

$$E(S, A) = \sum_{i=1}^{m} \sum_{x \in S_i} \|x - c_i\|^2, \tag{113}$$

where now $S_i = \left\{ x \in S | \, i = \arg\min_{j \in \{1,2,\ldots,m\}} \|x - c_i\| \right\}$. [9]

Different algorithms have been suggested to find the appropriate reference vectors c_i when their number m is given, for example k-means and LBG algorithms in [44] and [27]. Both methods assume, that the number of reference vectors is *a priori* given. The "right" number of clusters m remains an open question. This problem is considered, for example, in [9]. In this work, the iterative k-means algorithm is used to perform the vector quantization task. The procedure of this algorithm is given in the following:

k-Means Algorithm

Initialization: Arbitrarily choose m vectors as the initial set of code words in the codebook $A = \{c_1, c_2, \ldots, c_m\}$.

Nearest -Neighbour Search:
For $i = 1, 2, \ldots N$,
- Determine the closest representative, say c_j, for training vector $x_i \in S$.
- Set variable $b(i) = j$.
End$\{For\}$

Centroid Update:
For $j = 1, 2, \ldots m$,
- Determine c_j as the arithmetic mean of the vectors $x_i \in S$ with $b(i) = j$.
End$\{For\}$.

Iteration: Until no change in c_j's occurs between two successive iterations.

It was stated above, that the number of clusters has to be known in advance. Though given the liberty to initialize the codebook arbitrarily, it is known in practice, that this algorithm is highly sensitive to this factor. In this work, the Kohonen's self organizing map (SOM) is used to perform the initialization. The SOM is a widely used neural unsupervised clustering method and it gives a flexible method to use the training vectors themselves to initialize the codebook [19]. Here, the k-Means algorithm is allowed to make 100 iteration, at most.

After codebook generation we have a collection of reference vectors, which are assumed to be the representatives of the feature space S. These vectors are indexed by the integer set $0, 1, \ldots, m-1$, where m is the size of the codebook. For each input feature vector, the vector quantizer then finds the nearest reference vector, in Euclidean sense, from the codebook. After this the corresponding 1-dimensional index of the codebook becomes the "symbolic" representative of the multidimensional feature vector.

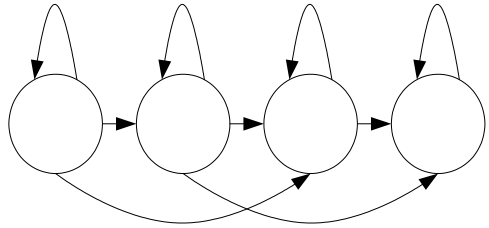

Figure 9. An illustration of a 4-state Bakis model.

Table 1. State number combinations.

Gesture	Comb. 1	Comb. 2	Comb. 3	Comb. 4
a	2	1	4	7
h, l	3	1	4	7
p, x	4	1	4	7
w,z	5	1	4	7

5.5 Choice of the model parameters

In section 1.4, the left-right topology was stated unanimously as the most natural choice for the topology of HMMs in the isolated hand gesture recognition. In example 7, it was presented a certain type of left-right Markov chain. In this work, it is used the Bakis type of topology of the underlying Markov chain. In a Bakis model state transitions to state itself, to the following and second following states are possible. This is illustrated in Figure 9.

For the consideration of the state number effects the number of states was varied in two ways:

1. more states for a more complicated gesture,

2. equal number of states for all models.

These ideas are tested with the combinations of the state numbers shown in Table 1.

The reasonability of the selection of the same amount of states for all models is confirmed in the works [23], [47] and [49]. In the first two works it

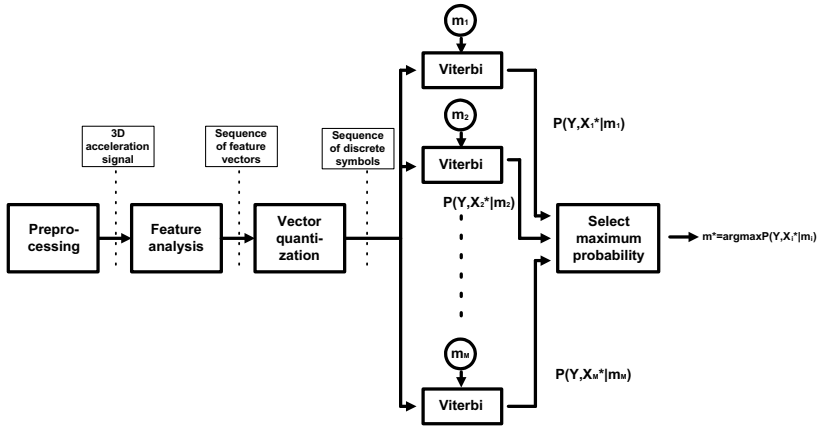

Figure 10. Block diagram of an isolated hand gesture HMM recognizer.

was used 5-state Bakis HMMs and in the third work, it was used 6-state Bakis models.

The size of the codebook implicitly orders the dimensionality of the state space of the observable process of a discrete HMM. Thus, the variation of the codebook size is equivalent to varying the number of possible observation symbols. The codebook size was given values 16, 32, 64, and 128. Once the feature extraction was performed, all 90 samples for each gesture were used to generate the codebook. The effects of varying the number of training sequences per model were investigated using 1, 5, 10, 20, 30, 50, 70, or 90 samples per model in the model estimation phase. For the testing of the recognition system it was used 50 samples per gesture. The overall structure of the used HMM recognizer is illustrated in Figure 10.

5.6 Implementation

The HMM recognition system was created offline in the MATLAB 5.3 environment. Preprocessing of the data was performed using ordinary functions available in MATLAB and Signal Processing and Statistics Toolboxes. The initialization of the codebook from a given training data was performed with the functions of the SOM toolbox, which is freely available in the Web.

A software package for Baum-Welch parameter estimation and Viterbi algorithm was implemented in a project on speech recognition with C++-language. As compiled programs, these algorithms were applied in the MATLAB environment, as well. [35]

6 Results

For fixed codebook sizes and four different state number combinations, it is presented recognition results (%) versus the number of used training sequences in Figures 11 - 18. Corresponding numerical values of the recognition rates are presented in Tables 2 - 9. The distributions of correct and false classifications for selected HMM recognizers are presented in confusion Tables 10 - 17.

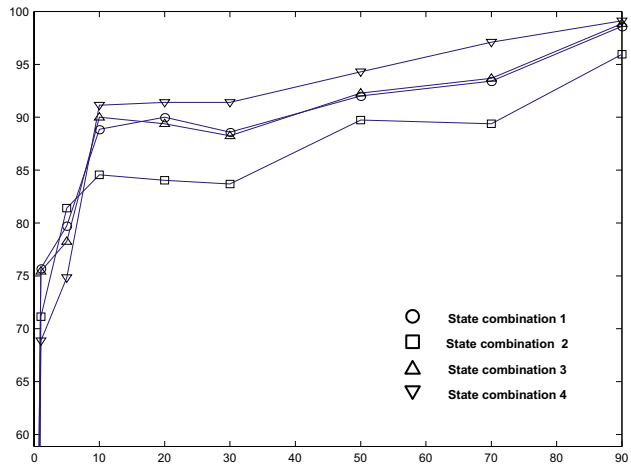

Figure 11. Recognition results (%) versus number of training sequences using codebook of size 16, four different state number combinations and feature 1.

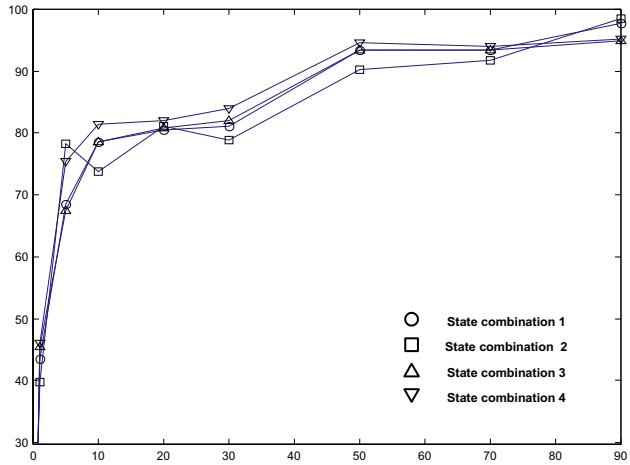

Figure 12. Recognition results (%) versus number of training sequences using codebook of size 32, four different state number combinations and feature 1.

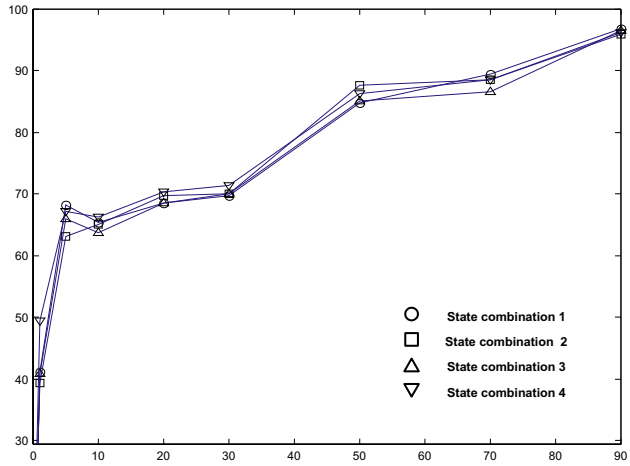

Figure 13. Recognition results (%) versus number of training sequences using codebook of size 64, four different state number combinations and feature 1.

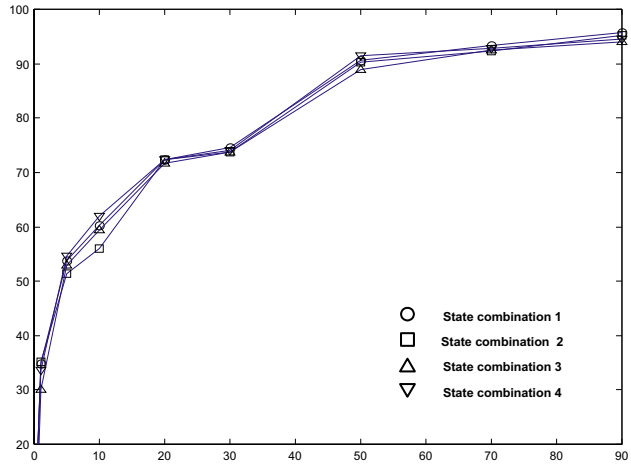

Figure 14. Recognition results (%) versus number of training sequences using codebook of size 128, four different state number combinations and feature 1.

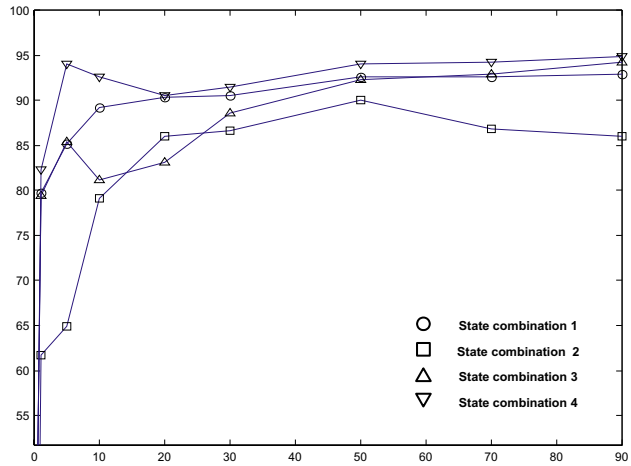

Figure 15. Recognition results (%) versus number of training sequences using codebook of size 16, four different state number combinations and feature 2.

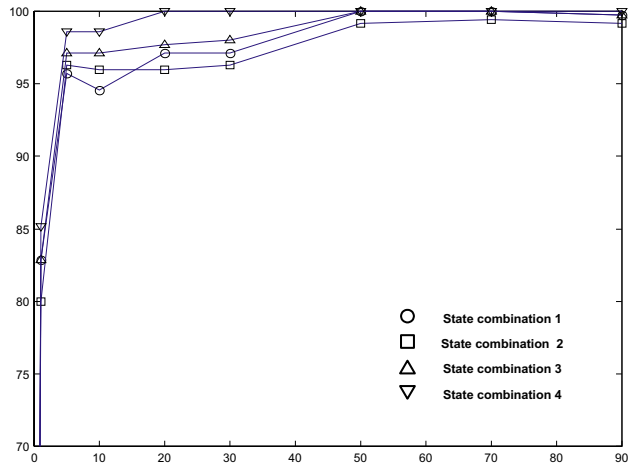

Figure 16. Recognition results (%) versus number of training sequences using codebook of size 32, four different state number combinations and feature 2.

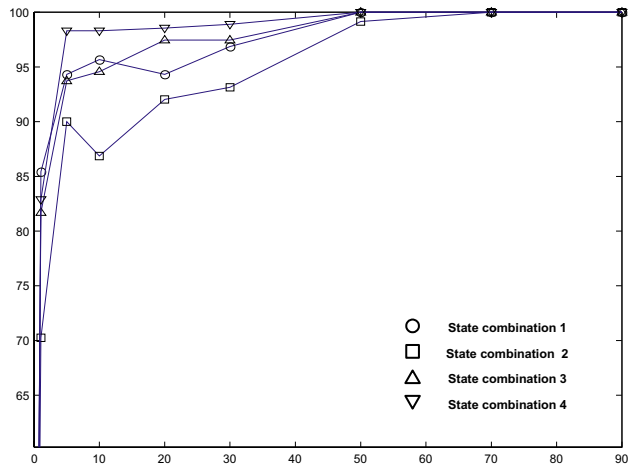

Figure 17. Recognition results (%) versus number of training sequences using codebook of size 64, four different state number combinations and feature 2.

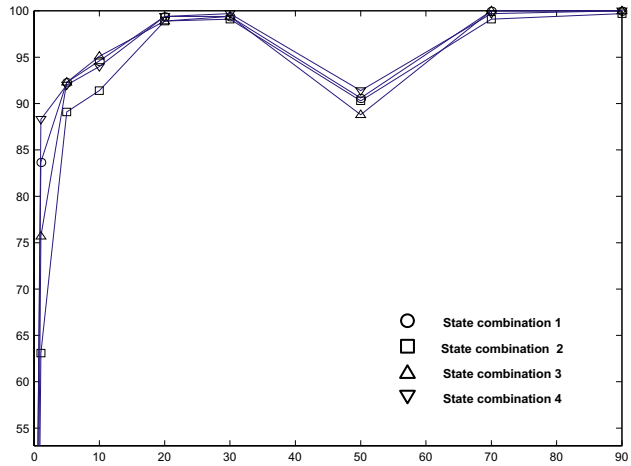

Figure 18. Recognition results (%) versus number of training sequences using codebook of size 128, four different state number combinations and feature 1.

Table 2. Total recognition results (%) using feature 1 and codebook of size 16. (Nsc = state number combination, Nts = number of training sequences)

Nsc\Nts	1	5	10	20	30	50	70	90
1	75.7	79.7	88.9	90.0	88.6	92.0	93.4	98.6
2	71.1	81.4	84.6	84.0	83.7	89.7	89.4	96.0
3	75.4	78.3	90.0	89.4	88.3	92.3	93.7	98.6
4	68.8	74.9	91.1	91.4	91.4	94.3	97.1	99.1

Table 3. Total recognition results (%) using feature 1 and codebook of size 32. (Nsc = state number combination, Nts = number of training sequences)

Nsc\Nts	1	5	10	20	30	50	70	90
1	43.4	68.6	78.6	80.6	81.1	93.4	93.4	97.7
2	39.7	78.3	73.7	81.1	78.6	90.3	91.7	98.6
3	45.4	67.4	78.6	80.6	82.0	93.4	93.4	94.9
4	46.0	75.4	81.4	82.0	84.0	94.6	94.0	95.1

Table 4. Total recognition results (%) using feature 1 and codebook of size 64. (Nsc = state number combination, Nts = number of training sequences)

Nsc\Nts	1	5	10	20	30	50	70	90
1	41.1	68.3	65.4	68.6	69.7	84.9	89.4	96.9
2	39.4	63.1	65.1	69.7	70.0	87.7	88.6	96.0
3	40.8	66.0	63.7	68.6	70.0	85.1	86.6	96.6
4	49.4	67.1	66.3	70.3	71.4	86.3	88.6	96.3

Table 5. Total recognition results (%) using feature 1 and codebook of size 128. (Nsc = state number combination, Nts = number of training sequences)

Nsc\Nts	1	5	10	20	30	50	70	90
1	34.9	53.7	60.3	72.3	74.6	90.6	93.4	95.7
2	35.1	51.4	56.0	72.3	73.7	90.3	92.3	96.0
3	30.0	52.9	59.4	71.7	73.7	88.9	92.6	94.0
4	33.7	54.6	62.0	72.3	74.0	91.4	92.9	94.6

Table 6. Total recognition results (%) using feature 2 and codebook of size 16. (Nsc = state number combination, Nts = number of training sequences)

Nsc\Nts	1	5	10	20	30	50	70	90
1	79.7	85.1	89.1	90.3	90.6	92.6	92.6	92.9
2	61.7	64.9	79.1	86.0	86.6	90.0	86.9	86.0
3	79.4	85.4	81.1	83.1	88.6	92.3	92.9	94.3
4	82.3	94.0	92.6	90.6	91.4	94.0	94.3	94.9

Table 7. Total recognition results (%) using feature 2 and codebook of size 32. (Nsc = state number combination, Nts = number of training sequences)

Nsc\Nts	1	5	10	20	30	50	70	90
1	82.9	95.7	94.6	97.1	97.1	100	100	99.7
2	80.0	96.3	96.0	96.0	96.3	99.1	99.4	99.1
3	82.9	97.1	97.1	97.7	98.0	100	100	99.7
4	85.1	98.6	98.6	100	100	100	100	100

Table 8. Total recognition results (%) using feature 2 and codebook of size 64. (Nsc = state number combination, Nts = number of training sequences)

Nsc\Nts	1	5	10	20	30	50	70	90
1	85.4	94.3	95.7	94.3	96.9	100	100	100
2	70.3	90.0	86.9	92.0	93.1	99.1	100	100
3	81.7	93.7	94.6	97.4	97.4	100	100	100
4	82.9	98.3	98.3	98.6	98.9	100	100	100

Table 9. Total recognition results (%) using feature 2 and codebook of size 128. (Nsc = state number combination, Nts = number of training sequences)

Nsc\Nts	1	5	10	20	30	50	70	90
1	83.7	92.3	94.6	99.4	99.4	90.6	100	100
2	63.1	89.1	91.4	98.9	99.1	90.3	99.1	99.7
3	75.7	92.3	95.1	98.9	99.4	88.9	100	100
4	88.3	92.0	94.0	99.4	99.7	91.4	99.7	100

Table 10. Confusion table of an HMM recognizer using feature 1, 50 training sequences and codebook of size 16.

47	0	0	2	1	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
0	0	0	50	0	0	0
0	0	3	0	34	13	0
0	0	0	0	0	50	0
0	0	0	0	0	1	49

Table 11. Confusion table of an HMM recognizer using feature 1, 90 training sequences and codebook of size 16.

49	0	0	1	0	0	0
0	48	0	0	2	0	0
0	0	50	0	0	0	0
0	0	0	50	0	0	0
0	0	0	0	50	0	0
0	0	0	0	0	50	0
0	0	0	0	0	0	50

Table 12. Confusion table of an HMM recognizer using feature 1, 50 training sequences and codebook of size 32.

32	0	0	16	0	0	2
0	50	0	0	0	0	0
0	0	49	0	0	0	1
0	0	0	50	0	0	0
0	0	0	0	50	0	0
0	0	0	0	0	50	0
0	0	0	0	0	0	50

Table 13. Confusion table of an HMM recognizer using feature 1, 90 training sequences and codebook of size 32.

35	0	0	14	0	0	1
0	50	0	0	0	0	0
0	0	48	0	0	1	1
0	0	0	50	0	0	0
0	5	0	0	45	0	0
0	0	0	0	0	50	0
0	0	0	0	0	0	50

Table 14. Confusion table of an HMM recognizer using feature 2, 5 training sequences and codebook of size 32.

50	0	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
0	0	0	50	0	0	0
0	5	0	0	45	0	0
0	0	0	0	0	50	0
0	0	0	0	0	0	50

Table 15. Confusion table of an HMM recognizer using feature 2, 90 training sequences and codebook of size 32.

50	0	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
0	0	0	50	0	0	0
0	0	0	0	50	0	0
0	0	0	0	0	50	0
0	0	0	0	0	0	50

Table 16. Confusion table of an HMM recognizer using feature 2, 5 training sequences and codebook of size 64.

50	0	0	0	0	0	0
0	45	1	0	4	0	0
0	0	50	0	0	0	0
0	0	0	50	0	0	0
0	5	0	0	45	0	0
0	0	0	0	0	50	0
0	0	1	0	0	0	49

Table 17. Confusion table of an HMM recognizer using feature 2, 90 training sequences and codebook of size 64.

50	0	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
0	0	0	50	0	0	0
0	0	0	0	50	0	0
0	0	0	0	0	50	0
0	0	0	0	0	0	50

7 Conclusions

The theoretical considerations in chapters 2 and 3 show the sound foundation of discrete HMMs. Mathematically simple structure of discrete Markov chains is adopted as a part of an HMM. Hidden Markov models give two useful possibilities of thinking its mathematical structure. One is the *doubly stochastic mathematical structure* consisting of the underlying hidden Markov chain and the observable process tied to the states of the hidden Markov chain. This illustrates the mechanism of creating an individual observation and by this way a whole sequence of observations from an HMM. From this point of view, it is found computationally feasible algorithms for the application of discrete HMMs: forward-backward algorithm for scoring, Viterbi algorithm for decoding and Baum-Welch algorithm for model parameter estimation. From more statistical point of view, HMMs may be seen as *piecewise stationary processes*, where each state transition leads to an output process of statistically fixed distribution of discrete symbols. Thus, modeling a signal with an HMM, assumes, that signal may be approximated as pieces with statistically unchangable properties in discrete time. Successive repetitions or returns to these pieces are restricted by the topology of the underlying Markov chain. A special type of the HMMs are the left-right HMMs. This type of HMMs are used to model signals whose properties change over time in a successive manner. The considerations of chapter 1 show, that this type of assumption is relevant with hand gestures.

For the study of hand gesture recognition it was created an isolated user-dependent hand gesture recognition system based on discrete hidden Markov models and Viterbi algorithm. A seven class classification problem was considered in this PR system, each class representing a predefined hand gesture. The three dimensional acceleration signal parametrisation with two different ways was tested in this system. Classification results in Tables 2 - 5 show , that by using feature 1, it is possible to create an HMM recognizer with recognition rate of at most 99.1 %. Feature 2 gives higher recognition rates, even of 100 %. This is illustrated by Figures 15 - 18. Feature 1 describes a hand gesture as a three dimensional signal, whose signal levels chance in a successive manner. These acceleration signal levels describe spatial orientation and acceleration of the input device. Feature 2 exploits the computation of more sophisticated entities of the acceleration signals. Parameter 1 represents the three dimensional signal level fluctua-

tion describing spatial orientation and and acceleration of the input device during the gesture. Parameters 2 and 3 reflect the dynamical changes during the hand gesture.

Codebook generation is an inevitable process, when discrete HMMs are used. A continuous multidimensional feature vector is converted into a discrete symbol, in this case, into a codebook index. The codebook was initialized with SOM clustering algorithm and further improved by the k-Means algorithm, since no automatic initialisation is included in k-Means algorithm. Figures 11 - 18, show that the need for bigger size of codebook grows when the dimensionality of the feature space grows. Codebook of size 16 was found the best among the chosen codebook sizes for feature 1, whereas codebooks sizes of 32 and 64 were found the best among the chosen codebook sizes for feature 2. These facts are can be seen in Figures 11 - 18. When HMMs are used, it is possible to think that two gestures are described as sequences of exactly same codebook vectors, but in different order. This kind of situation is created, when a small amount of clusters are expected to represent the feature space. This is not the optimal case. It is hoped for the codebook to have as many separate classes of vectors as possible for each gesture. The effect of too small codebook is clearly seen, in Figure 15, when feature 2 is used. On the other hand, the increment of the codebook size leads to a finer division of the feature space. This gives a possibility to describe the statistical variations of the gestures in more detail. Two feature vector sequences of the same gesture may be described as same vector sequences in a smaller codebook, whereas the increment of the codebook size, may split the two sequences into two partly separate sequences of codebook vectors. Consequently, the need of training sequences is increased to get a reliable knowledge of the statistical variation of the gestures in the chosen feature space.

In this work it was tested two ideas of selecting the number of states. The first idea was to give heuristically more states to a more complicated gesture, whereas the second idea was to give the same amount of states for all models. Giving all models the maximum seven states gave the most reliable recognizers within the chosen state number combinations and best codebook sizes. This is illustrated in Figures 11, 16 - 17. Giving more states to more complicated gestures was found a reasonable choice for the state number, also. It should be noted, that the definition of the complexity of a hand gesture is quite subjective a matter and not necessarily correct

with respect to the selected feature space. This kind of choice, may anyway be reasonable by the fact, that some gestures are short in time, whereas others may last twice the time. Giving a gesture more than or nearly the same amount of states as its length is as a sequence of feature vectors, is irrational. Keeping this in mind, the same amount of states for all models, pushes a shorter gesture ”go through” its states more quickly than the longer lasting gesture. One extreme was to give one state for all models. This is equivalent to look for the distribution of the symbols of the training sequences in the codebook. Tables 2 - 9 show, that results are good in this case. Though, the recognition rates grow slowly as a function of the amount of used training sequences. The good results may be explained by the small set of classes.

During the parameter estimation of the HMM models, it is ”shown” a collection of training sequences to the models. It is natural an assumption to expect better recognition results from an HMM recognizer, after having used more training sequences. This tendency is mainly observed in Figures 11 - 18 for both features. Though, the increment of the training sequences does not mean automatical improvement of the recognition results. Observed declines reflect the statistical variations of the training set, which can be seen in Figures 11 - 13 and 15 - 18. Less amount of training sequences sometimes model better the test set, than the incremental training set. The overall tendency shows greater reliability of the classifier, when the amount of training sequences is increased, anyway. The improvement of the recognizer is more slowly, when feature 1 is used. Mean while a fast improvement of the recognizer is seen, when feature 2 is used. This can be seen in Figures 16 - 17. It is also seen, that once radically good recognition results are achieved using feature 2, they are also maintained, when the ”experience” of the models is increased. This reflects, that feature 2 has more essential properties of the gestures, than gesture 1. Thus, gestures are well separated in the feature space generated by feature 2.

In above text, it is considered the total recognition rates of the HMM recognizers. Confusion Tables 10 - 17 are giving some examples, how the correct and false classifications are distributed in the seven classes. Tables 10 - 11 show, that classification errors may be due small amount of gestures, while the rest are purely classified. In Tables 10, 12 - 13, the first gesture is the main responsible for misclassifications. This is not notible, when feature 2 is used. Gesture 1 is the simplest and shortest in time. Therefore, the

relative representation of the feature vectors corresponding to gesture 1 is smaller than others. As a direct result of this, the relative representation for gesture 1 becomes smaller in the codebook, than for other gestures. Feature 1 is seen to be more sensitive to this temporal fact than feature 2.

References

[1] A. O. Allen, *Probability, Statistics, and Queueing Theory With Computer Science Applications*, Academic Press, Inc., 2nd edition, 1990.

[2] P. Baldi & S. Brunak, *Bioinformatics: The Machine Learning Approach*, The MIT Press, 1998.

[3] P. Baldi & Y. Chauvin, *Smooth On-Line Learning Algorithms for Hidden Markov Models*. Neural Computation, vol 6, pp. 307 - 318, 1994.

[4] L. E. Baum, *An Inequality and Associated Maximization Technique in Statistical Estimation for Probabilistic Functions of Markov Chains*, Inequalities, vol. 3, pp. 1 - 8, 1972.

[5] L. E. Baum & J. A. Eagon, *An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology*, Bulletin of the American Mathematical Society, vol. 73, pp. 360 - 363, 1967.

[6] L. E. Baum & T. Petrie, *Statistical Inference for Probabilistic Functions of Finite State Markov Chains,* Annals of Mathematical Statistics, vol. 37, pp. 1554 - 1563, 1966.

[7] L. E. Baum, T. Petrie, G. Soules & N. A. Weiss, *A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains*, Annals of Mathematical Statistics, vol. 41, pp. 164 - 171, 1970.

[8] L. E. Baum & G. R. Sell, *Growth Functions for Transformations on Manifolds*, Pacific Journal of Mathematics, vol. 27, pp. 211 - 227, 1968.

[9] H. Bischof, A. Leonardis & A. Selb, *MDL Principle for Robust Vector Quantization*, Pattern Analysis and Applications, Vol. 1, No. 2, pp. 59 - 72, 1999.

[10] K. L. Chung, *Markov Chains With Stationary Transition Probabilities*, Springer-Verlag, Heidelberg, 1967.

[11] J. R. Deller, J. G. Proakis & J. H. L. Hansen, Discrete-Time Processing of Speech Signals, Macmillan Publishing Company, Englewood Cliffs, pp. 677 - 744, 1993.

[12] A. El-Yacoubi, M. Gilloux, R. Sabourin & C. Y. Suen, An HMM-Based Approach for Off-Line Unconstrained Handwritten Word Modeling and Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, No. 8, pp. 752 - 760, August, 1999.

[13] Y. Ephraim, A. Dembo & L. R. Rabiner, Minimum Discrimination Information Approach for Hidden Markov Modeling, IEEE Transactions on Information Theory, Vol. 35, pp. 1000 - 1013, 1989.

[14] G. R. Grimmett & D. R. Stirzaker, Probability and Random Processes, Oxford University Press, Oxford, 1982.

[15] M. E. Harrington, R. W. Daniel & P. J. Kyberd, A Measurement System for the Recognition of Arm Gestures Using Accelerometers, Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, Vol. 209, No.2, pp. 129 - 133, 1995.

[16] T. Imielinski & H. F. Korth, Mobile Computing, Kluwer Academic Publishers, Boston, 1996.

[17] B. H. Juang & L. R. Rabiner, The Segmental K-means Algorithm for Estimating Parameters of Hidden Markov Models, IEEE Transactions on acoustics, Speech and Signal Processing, Vol 38, pp. 1639 - 1641, 1990.

[18] T. Kobayashi & S. Haruyama, Partly-Hidden Markov Model and Its Application to Gesture Recognition, ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings, Vol. 4, pp.3081-3084, 1997.

[19] T. Kohonen, Self-Organizing Maps, Springer Series in Information Sciences, Vol. 30, Springer, Berlin, Heidelberg, New York, 1995.

[20] T. Koski, A Course On Hidden Markov Models With application To Computational Biology (Lecture notes), Kungliga Tekniska Högskolan, 1999.

[21] E. Kreyszig, Advanced Engineering Mathematics, John Wiley & Sons, Inc., 7th edition, Singapore, 1993.

[22] S. Kullback, An Information-Theoretic Derivation of Certain Limit Relations for A Stationary Markov Chain, SIAM Journal of Control, pp. 454 - 459, 1966.

[23] C. Lee & Y. Xu, Online, Interactive Learning of Gestures for Human/Robot Interfaces, Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, 1996.

[24] H.-K. Lee & J.-H. Kim, Gesture Spotting from Continuous Hand Motion, Pattern Recognition Letters, Vol. 19, pp. 513 - 520, 1998.

[25] S. E. Levinson, L. R. Rabiner & M. M. Sondhi, An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition, The Bell System Technical Journal, Vol. 62, No 4, pp. 1035 - 1074, 1983.

[26] S. E. Levinson, L. R. Rabiner & M. M. Sondhi, On the Application of Vector Quantization and Hidden Markov Models to Speaker-Independent, Isolated Word Recognition. Bell System Technical Journal, vol. 62, no. 62, pp. 1075 - 1105, Apr. 1983.

[27] Y. Linde, A. Buzo & R. M. Gray, An Algorithm for Vector Quantizer Design, IEEE Transactions on Communications, Vol. 28, pp. 84 - 95, 1980.

[28] I. L. MacDonald & W Zucchini, Hidden Markov and Other Models for Discrete-valued Time Series, Chapman & Hall, Monographs on statistics and Applied Probability 70, London, Weinheim, New York, Melbourne, Madras, 1997.

[29] J. Martin & J.-B. Durand, Automatic Handwriting Gestures Recognition Using hidden Markov Models, Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition, pp. 403 - 409, 2000.

[30] S. J. McKenna & S. Gong, Gesture Recognition for Visually Mediated Interaction Using Probabilistic Event Trajectories, Proceedings of the Ninth British Machine Vision, pp. 498 - 507, 1998.

[31] G. J. McLachlan & T. Krishnan, The EM Algorithm and Extensions, John Wiley & Sons, New York, pp. 35 - 37, 233 - 237, 1997.

[32] B.-W. Min, H.-S. Yoon, J. Soh, Y.-M. Yang & T. Ejima, Hand gesture recognition using hidden Markov models, IEEE International Conference on Systems, Man, and Cybernetics, Vol. 5, pp. 4232 - 4235, 1997.

[33] J. R. Norris, Markov Chains, Cambridge University Press, Cambridge, 1997.

[34] V. I. Pavlovic, R. Sharma & T. S. Huang, Visual Interpretation of Hand Gestures for Human-Computer Interaction: A review, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, July 1997.

[35] J. Peltola, HMM-perustainen puheentunnistus. Diploma thesis, in Finnish, University of Oulu, 1998.

[36] L. R. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE, Vol. 77, pp. 257 - 286, 1989.

[37] L. R. Rabiner & B. H. Juang, An Introduction to Hidden Markov Models, IEEE ASSP Magazine, Vol. 3, No. 1, January, 1986.

[38] L. R. Rabiner & B. H. Juang, Fundamentals of Speech Recognition, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[39] R. T. Rockafellar, Convex Analysis, Princeton University Press, Princeton, New Jersey, 1970.

[40] S. M. Ross, Introduction to Probability Models, 4th edition, Academic Press, Orlando, 1989.

[41] H. Sawada & S. Hashimoto, Gesture recognition using an acceleration sensor and its application to musical performance control, Electronics and Communications in Japan, Part 3, Vol. 80, No. 5, 1997.

[42] R. Schallkoff, Pattern Recognition: Statistical, Syntactic and Neural Approaches, John Wiley & Sons, New York, 1992.

[43] P. Smyth, Hidden Markov Models for Fault Detection in Dynamic Systems, Pattern Recognition, Vol. 27, No. 1, pp. 149 - 164, 1994.

[44] S. Theodoridis & K. Koutroumbas, Pattern Recognition, Academic Press, New York, 1999.P.

[45] P. Tuominen & P. Norlamo, Todennäköisyyslaskenta osat 1-2, Limes ry, Helsinki, 1974.

[46] W. Turin & M. M. Sondhi, Modeling Error Sources in Digital Channels, IEEE Journal on Selected Areas in Communications, Vol. 11, No. 3, April, 1993.

[47] S. Usa & Y. Mochida, A conducting recognition system on the model of musicians' process, The Journal of the Acoustical Society of Japan (E), Vol. 19, No. 4, 1998.

[48] A. J. Viterbi, Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence, IT-3, pp. 260 - 269, 1967.

[49] J. Yang, Y. Xu & C. S. Chen, Gesture Interface: Modeling and Learning, Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 2, pp. 1747 - 1752, 1994.

Published by

Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
Phone internat. +358 9 4561
Fax +358 9 456 4374

Series title, number and report code of publication

VTT Publications 449
VTT–PUBS–449

Author(s)
Mäntylä, Vesa-Matti

Title

Discrete hidden Markov models with application to isolated user-dependent hand gesture recognition

Abstract

The development of computers and the theory of doubly stochastic processes, have led to a wide variety of applications of the hidden Markov models (HMMs). Due to their computational efficiency, discrete HMMs are often favoured. HMMs offer a flexible way of presenting events with temporal and dynamical variations. Both of these matters are present in hand gestures, which are of increasing interest in the research of human-computer interaction (HCI) technologies. The exploitation of human-to-human communication modalities has become actual in HCI applications. It is even expected, that the existing HCI techniques become a bottleneck in the effective utilization of the available information flow.

In this work it is given mathematically uniform presentation of the theory of discrete hidden Markov models. Especially, three basic problems, scoring, decoding and estimation, are considered. To solve these problems it is presented forward and backward algorithms, Viterbi algorithm, and Baum-Welch algorithms, respectively.

The second purpose of this work is to present an application of discrete HMMs to recognize a collection of hand gestures from measured acceleration signals. In pattern recognition terms, it is created an isolated user-dependent recognition system. In the light of recognition results, the effect of several matters to the optimality of the recognizer is analyzed.

Keywords
Discrete hidden Markov models, hand gesture recognition, stochastic processes, discrete Markov chains, Bayes classification

Activity unit
VTT Electronics, Networking Research, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland

ISBN	Project number
951–38–5875–8 (soft back ed.) 951–38–5876–6 (URL:http://www.inf.vtt.fi/pdf/)	E6SU00081

Date	Language	Pages	Price
October 2001	English	104 p.	C

Name of project	Commissioned by
ITEA 99002 BEYOND	Tekes

Series title and ISSN	Sold by
VTT Publications 1235–0621 (soft back ed.) 1455–0849 (URL: http://www.inf.vtt.fi/pdf/)	VTT Information Service P.O.Box 2000, FIN-02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374