VILNIUS UNIVERSITY

Jūratė
VAIČIULYTĖ

Study and Application of Hidden Markov Models to Online Analysis of Multivariate Sequence Data

SUMMARY OF DOCTORAL DISSERTATION

Natural Sciences,
informatics N 009

VILNIUS 2020

This dissertation was written between 2015 and 2019 at Vilnius University.

Academic supervisor:
Prof. Habil. Dr. Leonidas Sakalauskas (Vilnius University, Natural Sciences, Informatics – N 009).
This doctoral dissertation will be defended in a public meeting of the Dissertation Defence Panel:
Chairman:
Prof. Dr. Olga Kurasova (Vilnius University, Natural Sciences, Informatics – N 009).

Members:
Prof. Dr. Stefano Bonnini (University of Ferrara, Italy, Natural Sciences, Informatics – N 009);
Prof. Dr. Rytis Maskeliūnas (Kaunas University of Technology, Technological Sciences, Informatics Engineering – T 007);
Assoc. Prof. Gintautas Tamulevičius (Vilnius University, Technological Sciences, Informatics Engineering – T 007);
Prof. Dr. Julius Žilinskas (Vilnius University, Natural Sciences, Informatics – N 009).

The dissertation shall be defended at a public meeting of the Dissertation Defence Panel at 12:00 p. m. on 23rd of June, 2020 in Room 203 of the Institute of Data Science and Digital Technologies of Vilnius University.
Address: Akademijos str. 4, LT-04812 Vilnius, Lithuania.
The summary of the doctoral dissertation was distributed on the 22nd of May 2020.
The text of this dissertation can be accessed at the library of Vilnius University, as well as on the website of Vilnius University: *www.vu.lt/lt/naujienos/ivykiu-kalendorius*

VILNIAUS UNIVERSITETAS

Jūratė
VAIČIULYTĖ

Paslėptųjų Markovo modelių tyrimas ir taikymas daugiamačių sekų palaipsnei analizei

DAKTARO DISERTACIJOS SANTRAUKA

Gamtos mokslai,
Informatika N 009

VILNIUS 2020

Disertacija rengta 2015–2019 metais Vilniaus universitete.

Mokslinis vadovas:
prof. habil. dr. Leonidas Sakalauskas (Vilniaus universitetas, gamtos mokslai, informatika – N 009).

Gynimo taryba:
Pirmininkė – prof. dr. Olga Kurasova (Vilniaus universitetas, gamtos mokslai, informatika – N 009).

Nariai:
prof. dr. Stefano Bonnini (Feraros universitetas, Italija, gamtos mokslai, informatika – N 009);
prof. dr. Rytis Maskeliūnas (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – T 007);
doc. dr. Gintautas Tamulevičius (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – T 007);
prof. dr. Julius Žilinskas (Vilniaus universitetas, gamtos mokslai, informatika – N 009).

Disertacija ginama viešame Gynimo tarybos posėdyje 2020 m. birželio mėn. 23 d. 12:00 val. Vilniaus universiteto Duomenų mokslo ir skaitmeninių technologijų instituto 203 auditorijoje.
Adresas: Akademijos g. 4, LT-04812 Vilnius, Lietuva.
Disertacijos santrauka išsiuntinėta 2020 m. gegužės mėn. 22 d.

Disertaciją galima peržiūrėti Vilniaus universiteto bibliotekoje ir VU interneto svetainėje adresu: *https://www.vu.lt/naujienos/ivykiu-kalendorius*

1 INTRODUCTION

1.1 Research area and relevance of the problem

An important feature of various applications (such as computer vision, speech recognition, image analysis, etc.) is the large and continuously streaming data. They can be processed quite efficiently in real time using various algorithms. Thus, machine learning algorithms are under improvement due to new and emerging computational challenges of real-time data interpretation, learning, and decision making. It is difficult to apply traditional deep learning methods in areas of this nature, where data needs to be processed and used for training in real-time, because they require a large static training dataset and sufficient computational resources. However, attempts are being made to model these large data streams of emerging real-time systems as stochastic processes. One of widely applied modelling approaches of stochastic processes is hidden Markov models (HMM).

In recent decades, a modified task for the HMM parameters estimation has been raised. It introduces an additional requirement that observations must be processed online (i.e., incrementally, recursively) rather than stored in computer memory and processed as a single set. This formulation of the online estimation of HMM parameters has become of great theoretical and practical significance, as in some applications it is not computationally possible to store and process large batches of observations. Online estimation of HMM parameters is also important in applications where HMM parameters may change over time.

The Baum-Welch (batch) algorithm is a popular and effective HMM parameter estimation algorithm. It inspired the development of many online HMM parameter estimation algorithms [1–5]. Other proposed methods for estimating HMM parameters are based on recursive maximum likelihood methods [6, 7], prediction error methods [7–9].

However, these online HMM parameter estimation techniques tend to converge to local (non-global) extremes of their target functions.

Recently, several new methods (with proven convergence properties) using ergodic (hidden) Markov chain state processes and information theories have been proposed for HMM state transition parameters' estimation for one-dimensional Gaussian models [10, 11]. These evaluation methods are consistent in estimating the HMM state transition parameters under thing full knowledge of the HMM process. In the case of unknown HMM parameters, consistent estimation of multivariate (Gaussian or non-Gaussian) HMM parameters is still a significant unresolved task. This dissertation presents a new methodology for developing the equivalent online (recursive) version of batch algorithms.

1.2 Research object

The research object of the dissertation is an online parameter estimation algorithms of discrete-state hidden Markov models when the observations are multivariate continuous.

1.3 The aim and objectives of the research

Aim:

- Development, study and application of online discrete-state hidden Markov model multivariate parameter estimation algorithms in observation sequence analysis.

Objectives:

- To analyse the hidden Markov model parameter estimation algorithms in real-time observation sequence analysis.
- To create online hidden Markov model multivariate parameter estimation algorithms.
- To perform experiments by statistical modelling in order to prove that the proposed algorithm is effective in online parameter

estimation and compare it with other HMM parameter estimation algorithms.
- To apply proposed algorithms to observation sequence analysis.

1.4 Research methodology

In order to achieve the aim of the work, the scientific works in the field of the online HMM parameter estimation were analysed. Methods of information search, systematization, analysis, comparative analysis and generalization were used. The investigation of the developed online HMM parameter estimation algorithm by the Monte-Carlo method and solving test problems conducted. The processing of observations and statistical analysis of experimental results was performed; the method of comparison and generalization was used to evaluate the results obtained. Algorithm theories, data mining, statistical analysis, and recognition theory knowledge was used in the work.

1.5 Scientific novelty

- Online HMM parameter estimation algorithms in sequential analysis of multivariate observations developed and experimentally investigated.
- Recursive calculation of state transition probabilities of multivariate Gaussian hidden Markov models based on the Chapman-Kolmogorov equation proposed. This method allows achieving higher classification accuracy than using the forward procedure used in known online algorithms.
- Online HMM parameter estimation algorithm when the observations are distributed according to the Dirichlet distribution suggested.
- Proposed algorithms applied in isolated word recognition, occupancy detection and pulsar identification problems. The experimental results confirmed the efficiency of the proposed

algorithms – reduced calculation time and slightly changed (up to 3%) classification accuracy.
- Research of online algorithms can be applied in the development of online equivalents of other types of algorithms for real-time classification.

1.6 Practical novelty of the work

The proposed online estimation of multivariate HMM parameters can be used in various multivariate data processing systems and tools in which the analysed data is of stochastic origin. Data storage requirements are limited.

- Online Gaussian HMM parameter estimation algorithm was applied to isolated word recognition when a fixed amount of speech data is for training, and further speech data is classified and used for parameter re-estimation.
- Online Dirichlet HMM parameter estimation algorithm was applied to the occupancy detection problem.
- Online Dirichlet HMM parameter estimation algorithm was applied to the pulsar identification problem.

1.7 Statements to be defended

- The online Gaussian HMM parameters estimation algorithm has linear time complexity and is equivalent to the batch Baum-Welch algorithm in terms of classification accuracy.
- The application of the Chapman-Kolmogorov equation in the calculation of the state transition probabilities improves the convergence of the model parameters compared to the traditional forward procedure.
- Sufficient dataset for the initial approximation of the algorithm, which ensures the stability of the algorithm exists and prevents

it from converging to the distorted local extremes of the likelihood function.

- Online Gaussian and Dirichlet HMM parameter estimation algorithms can be applied to solve practical classification problems, when the process observed can be modelled by hidden Markov models.

1.8 Outline of the dissertation

The dissertation consists of five chapters and a list of references. The chapters of the dissertation are as follows: Introduction; Review of online HMM parameter estimation methods; Online Gaussian HMM parameter estimation; Online Dirichlet HMM parameter estimation; Applications; Conclusions. This work contains 128 pages that include 23 figures and 29 tables; the list of references consists of 120 sources.

2 REVIEW OF ONLINE HMM PARAMETER ESTIMATION METHODS

The Hidden Markov Model (HMM) is a widespread tool for modelling stochastic processes. HMMs have been applied in statistics [12], machine learning [13], signal and image processing [14], signal tracking [15], prediction [16], and others. HMMs are Markov chains where observation is a random state function. HMM consists of two stochastic processes where a hidden process can be monitored only through a sequence of observations generated by other stochastic processes.

The considered HMM is discrete-state and observations are assumed to be continuous. The HMM parameters are the state transition and output probabilities. Given the values of the hidden variable s, a conditional probability distribution of the hidden variable $s(t)$ at time t depends only on the value of the hidden variable $s(t-1)$. Additionally, the value of the observed variable $O(t)$ depends only on the value of the hidden variable $s(t)$. A set of output probabilities is assigned for each of the N-possible states. The probability calculation considers the state of the hidden variable and reflects the observed variable's distribution at a given point in time.

The Markov process (hidden under the double line in Fig. 1) is determined by the current state and the transition probability matrix A. Observable O_i values are related to the hidden states of the Markov process through the probability density function B [17]. The arrows in Figure 1 show the relative dependencies, and the X_i values represent the hidden state sequence.

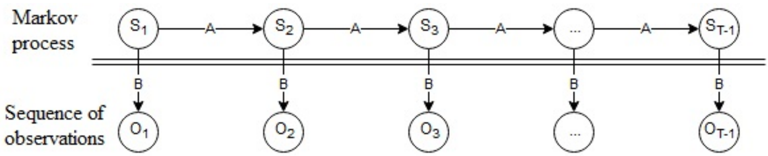

Figure 1: Hidden Markov model

HMM is defined by the set $\lambda = (\mathbf{A}, \mathbf{B}, \pi)$ [18]. When training an HMM, the model parameters estimated are the transition from one state to another state probability matrix and the observation probability density function for each model state. Probability distribution can take continuous, normal or other forms.

The standard methods for estimating HMM parameters are based on batch learning using expectation-maximization (EM) methods [5] such as the Baum-Welch algorithm [2] or numerical optimization methods. In such cases, the HMM parameters are estimated over several training iterations until the objective function (e.g., the maximum likelihood function) is maximized. For a batch algorithm, a fixed size set of training observations is assumed to be available for estimation. Given a new training dataset, a batch algorithm cannot apply a new dataset to HMM without being retrained with the aggregated data.

As of late, much consideration has been paid to HMM parameters estimation when a small training observation set is given and further re-estimation is performed with observations obtained sequentially over time [3, 19]. These online algorithms do not store observations in the computer's memory. They use the results of prior observations to evaluate the parameters when they are received. These methods differ from batch HMM parameter estimation methods which process blocks of observations only after they have been stored in memory.

Online HMM training methods can be divided into three categories: standard numerical optimization, expectation–maximization, and recursive estimation [19]. The most popular online (and offline) HMM parameter estimation technique is based on the maximum likelihood method (MLM). Modifications of the MLM that apply different numerical optimization techniques have been proposed in various papers [1, 5, 18–20].

As an alternative to online HMM parameter estimation by EM, some authors have proposed recursive algorithms for parameter estimation based on the maximum likelihood method, in which the likelihood function is optimized using stochastic gradient methods [6, 7, 21], Newton methods [14], or others [22, 23]. Most of the proposed methods reflect the offline Baum-Welch algorithm well, but their stability and convergence properties are poorly explored.

In many existing papers, the authors examined only the general theoretical features of probabilistic models [24], the algorithms are largely schematic [1, 19, 25], and the numerical experiments are presented with one-dimensional data [1, 3, 26]. Conceptual diagrams of online algorithms are presented but the convergence of these algorithms is not studied, and the value of such algorithms remains unclear [27].

A novel online HMM parameter estimation algorithms are developed and implemented in this work that focuses mainly on continuous HMM. The unknown model parameters are estimated by the maximum likelihood method.

3 ONLINE GAUSSIAN HMM PARAMETER ESTIMATION

In the considered discrete-state HMM, observations are assumed to be continuous and to follow Gaussian distribution. HMM parameters are state transition probabilities and output probabilities.

The parameters of an implemented HMM mathematical model are as follows:

- T – the length of the observation sequence,
- N – the number of states in the model,
- **A** – the state transition probability matrix $\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix}$,
- the initial state distribution vector $\pi = \begin{bmatrix} \pi_1 & \cdots & \pi_N \end{bmatrix}^T$,
- and the probability density function (PDF)

$$N\left(\mu_s, \sigma_s\right) = \frac{1}{\sqrt{\left(2\pi\right)^M \left|\sigma_s\right|}} e^{-\frac{1}{2}\left(\mathbf{o}-\mu_s\right)^T \sigma_s^{-1} \left(\mathbf{o}-\mu_s\right)}. \tag{1}$$

Observations are defined by a normal distribution with a mean of μ_s and a covariance of σ_s, $1 \leq S \leq N$:

$$\mu_s = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_M \end{bmatrix}, \sigma_s = \begin{bmatrix} \sigma_{11} & \cdots & \sigma_{1M} \\ \vdots & \ddots & \vdots \\ \sigma_{M1} & \cdots & \sigma_{MM} \end{bmatrix},$$

where M is the number of elements.

For the sake of simplicity, we assume that the distribution of each state is normal. The parameters μ and σ for the states S_i and S_j are different when $i \neq j$. In other words, if there are several states in the chain where the observed signals have the same parameters (μ and σ), they can be combined into a single state.

HMM parameters can be re-estimated iteratively using the maximum likelihood method. We will discuss the maximum likelihood method in case of continuous random variables.

Suppose we observe a random variable $\mathbf{o}$, whose density $b(\mathbf{o})$ depends on unknown parameters. The probability density of the observation is defined as $\sum_{j=1}^{N} \pi_j b_j(\mathbf{o})$, where π_j is the probability of being in state j, and the probability of an observation in state j is a multivariate Gaussian: $b_j(\mathbf{o}) = N(\mu_j, \sigma_j)$. We can define the log-likelihood function of the observation which has a probability π being in a state: $l(\mathbf{o}, \mu, \sigma, \pi) = \frac{(\mathbf{o}-\mu)^T \sigma^{-1} (\mathbf{o}-\mu)}{2} + \ln\left(\frac{\sqrt{|\sigma|}}{\pi}\right)$, First, we define
$L(\mathbf{o}, \mu, \sigma, \pi) = \sum_{i=1}^{N} e^{-l(\mathbf{o}, \mu_i, \sigma_i, \pi)}$.
Distribution parameter estimates have to maximize log-likelihood function:

$$l(\mathbf{o}, \mu, \sigma, \pi) \to \max_{\mathbf{o}, \mu, \sigma, \pi}. \quad (2)$$

Then, we calculate $\ln L$ derivatives according to μ and σ:
$(L_{\mu_i})' = \frac{e^{-l(\mathbf{o}, \mu_i, \sigma_i, \pi)} (\mathbf{o}-\mu_i) \sigma_i^{-1}}{\sum_i e^{-l(\mathbf{o}, \mu_i, \sigma_i, \pi)}}$,
$(L_{\sigma_i})' = \frac{e^{-l(\mathbf{o}, \mu_i, \sigma_i, \pi)} \left(\sigma^{-1} (\mathbf{o}-\mu)(\mathbf{o}-\mu)^T \sigma^{-1} - \sigma^{-1}\right)}{\sum_i e^{-l(\mathbf{o}, \mu_i, \sigma_i, \pi)}}$.

Finally, the equations with the derivatives $(\ln L_{\mu_i})' = 0$ and $(\ln L_{\sigma_i})' = 0$ are solved with respect to μ and σ.

In case of multivariate normal distribution, the standard EM algorithm is used to maximize the mean and variance estimates of the HMM model. The EM algorithm involves the following steps [28]:

- E step: calculate the log-likelihood function $L(\theta^i) = E\left[logL\left(S^i|\theta\right)\right]$.
- M-step: maximize the conditional mean of the likelihood function $\theta^{i+1} \to \max_\theta L(\theta^i)$.

Then, the HMM parameter re-estimation formulas through sums are as follows:

$$\bar{\mu_j} = \frac{\sum_{t=1}^{T} (\gamma_t(j) \cdot \mathbf{o}_t)}{\sum_{t=1}^{T} \gamma_t(j)}, \quad (3)$$

$$\bar{\sigma_j} = \frac{\sum_{t=1}^{T} \gamma_t(j) \cdot (\mathbf{o}_t - \bar{\mu_j})(\mathbf{o}_t - \bar{\mu_j})^T}{\sum_{t=1}^{T} \gamma_t(j)}, \quad (4)$$

$$\gamma_t(j) = \frac{e^{-l(\mathbf{o}_t, \bar{\mu_j}, \bar{\sigma_j}, \pi)}}{\sum_{i=1}^{N} e^{-l(\mathbf{o}_t, \bar{\mu_i}, \bar{\sigma_i}, \pi)}}, \quad (5)$$

$$\bar{a_{ij}} = \frac{\text{expected number of transitions from } S_i \text{to } S_j}{\text{expected number of transitions from } S_i}. \quad (6)$$

Then, $\bar{a_{ij}}$ is estimated using the Forward-Backward algorithm [18].

The batch EM algorithm has quadratic running time. Obtaining the maximum likelihood estimates will require calculations that are proportional to the fixed sample size. If the re-estimation is performed for each observation, the total number of required operations will have quadratic time complexity – the number of operations required for re-estimation will grow significantly, and real-time re-estimation will become impossible. To solve this problem, recursive formulas for the HMM parameter re-estimation were derived. They do not have infinite sums and are expressed in terms of averages that converge to finite values as the number of iterations increase. In this case, the HMM parameters are updated with each new observation received without storing the previous observations.

Besides, it is generally hard to implement the forward-backward procedure in online algorithms. Therefore, the backward procedure is often skipped or other smoothing methods are applied to calculate the state transition probabilities. We propose to calculate state transition probabilities using the Chapman-Kolmogorov equation, which calculates the transition probability for system to be in a state at time t, if at time $t-1$ it was in state i [29,30]: $\pi_t = \mathbf{A} \cdot \pi_{t-1}$.

We denote the state transition probabilities as $\beta_t^i = \frac{1}{t} \sum_{i=1}^{t} \omega_t^i$, where the coefficients are $\omega_t^i = \frac{e^{-l(\mathbf{o}_t, \hat{\mu_i}, \hat{\sigma_i}, \pi_i)}}{\gamma_t}$, $\gamma_t = \sum_{i=1}^{N} e^{-l(\mathbf{o}_t, \hat{\mu_i}, \hat{\sigma_i}, \pi_i)}$.

Proposition. The following recursive formulas can be used to re-estimate the mean vector and covariance matrix in (3) and (4):

$$\mu_t^i = \mu_{t-1}^i + \frac{\left(\mathbf{o}_t - \mu_{t-1}^i\right)}{t} \cdot \frac{\omega_t^i}{\beta_t^i}, \tag{7}$$

$$\sigma_t^i = \frac{\beta_{t-1}^i \cdot (t-1)}{\beta_t^i \cdot t} \left(\sigma_{t-1}^i + \frac{\left(\mathbf{o}_t - \mu_{t-1}^i\right)\left(\mathbf{o}_t - \mu_{t-1}^i\right)^T}{t} \cdot \frac{\omega_t^i}{\beta_t^i} \right), \tag{8}$$

$$\beta_t^i = \beta_{t-1}^i + \frac{1}{t}\left(\omega_t^i + \beta_{t-1}^i\right). \tag{9}$$

Proof. It is easy to verify that $\beta_t^i = \beta_{t-1}^i + \frac{1}{t}\left(\omega_t^i + \beta_{t-1}^i\right) = \frac{1}{t-1}\sum_{i=1}^{t-1}\omega_t^i + \frac{1}{t}\left(\omega_t^i - \frac{1}{t-1}\sum_{i=1}^{t-1}\omega_t^i\right) = \frac{1}{t}\sum_{i=1}^{t}\omega_t^i$. Similarly formulas (7) and (8) follow from (3) and (4), respectively.

We can apply formulas (7–9) starting from some initial approximation. The proposed online algorithm – Online Gaussian Hidden Markov Model Parameter Estimation Algorithm (O_GHMM_PEA) – has two main parts. The first part is model training used for initial parameter evaluation with a given small fixed-size set of training observations. When the initial parameter estimates are obtained, further identification and re-estimation of parameters are carried out by observing the process in real time. Initial estimates of the online algorithm are essential to ensure its consistency and to prevent convergence to distorted local extremes of the likelihood function.

It is easy to see that in case of sequential observation analysis, the time complexity of the proposed online algorithm O_GHMM_PEA is $O(n)$.

The algorithm stops when a standard error is less or equal to the ratio Δ: $\bar{L} = \frac{\sum_{i=1}^{N} L_i}{N}$, $D^2 = \frac{1}{N}\sum_{i=1}^{N}(L_i - \bar{L})^2$, then $\Delta \sim \frac{\sqrt{D}}{\sqrt{N}} \leq \epsilon$.

3.1 Computational results

The computational results presented below were obtained from the implementation of the batch EM and O_GHMM_PEA algorithms.

The results showing the performance of the O_GHMM_PEA algorithm were obtained by applying it to the clustered data. The clustering datasets were generated randomly from two ($N = 2$) multivariate normal distributions with the following characteristics:

- Centroid 1: $[500,\ 500,\ \ldots]$, Centroid 2: $[600,\ 600,\ \ldots]$;
- Dimensions: 2, 4, 8 and 16;
- Standard deviation: 20, 30, 40 and 50;
- State transition probability matrix: $A = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$;
- Initial state probability distribution vector: $\pi = [0.5\ \ 0.5]^T$;

The initial estimation of the parameters was carried out with the O_GHMM_PEA algorithm using a small set of training observations. The parameters of the model were randomly initialised. A total of 100 experiment replicates were carried out, and the results were averaged. In all experiments the termination criterion was set to $\epsilon = 0.01$.

3.1.1 Experimental results for computational time

Table 1 presents the computational time results for the clustering dataset with O_GHMM_PEA and, for comparison, the batch EM algorithm. The columns show the number of observations (T), the dimensions of the observation ($M = 2, 4, 8, 16$), CPU time (Time) in seconds of the two algorithms (O_GHMM_PEA (Rec.) and batch EM (Batch)), and time ratio (Ratio).

The average computational time (in seconds) of the algorithm was measured with over 100 experiments using the generated dataset. A strong speed advantage over the batch EM algorithm was achieved with the O_GHMM_PEA. The computational time of the O_GHMM_PEA is 3–9 times faster than that of the batch EM algorithm even if the dimensions of observations increase.

Table 1: Computational times of the implemented algorithms

T=	M = 2		M = 4		M = 8		M = 16		Ratio
	Rec.	Batch	Rec.	Batch	Rec.	Batch	Rec.	Batch	
1000	0.08	0.26	0.11	0.27	0.18	0.55	0.38	0.95	0.37
3000	0.13	0.8	0.17	0.79	0.28	1.5	0.58	2.66	0.20
5000	0.19	1.46	0.24	1.41	0.4	2.62	0.79	4.46	0.16
7000	0.24	2.15	0.31	2.02	0.51	3.77	0.98	6.37	0.14
10000	0.33	3.27	0.41	3.12	0.63	5.26	1.29	9.11	0.13

3.1.2 Experiments for algorithm convergence property

The standard error of the estimated HMM model parameters as the difference between the real values of the parameters and the estimated model parameters was determined as follows in order to analyse how close the trained parameter values of the O_GHMM_PEA converge to the original parameter values Standard error $= \frac{1}{NM}\sum(x-\hat{x})'(x-\hat{x})$.

Figure 2 indicates that the O_GHMM_PEA and the batch EM algorithm converge to the same value. In terms of convergence the O_GHMM_PEA does not significantly deviate from the batch EM algorithm. When more training data are available, both the O_GHMM_PEA and the batch EM algorithms consistently improve recognition accuracy. The spread of the data and the number of dimensions influence the evaluation of model parameters. However, the slight difference between the two algorithms' results suggests that the O_GHMM_PEA is successful in learning the model parameters. Figure 3 shows the ratio of standard errors of the O_GHMM_PEA and batch EM algorithms. The difference between average standard error of O_GHMM_PEA and batch EM algorithm is 3%.

The results of the state transition probability matrix calculation obtained during parameter estimation of the batch EM and O_GHMM_PEA algorithms are presented in Table 2, which shows the dataset size (T), the dimensions of the observations (M), and the

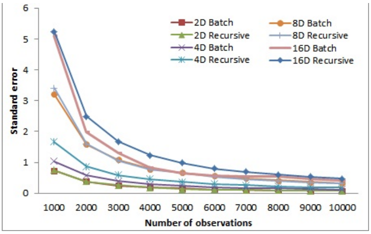

Figure 2: SE of O_GHMM_PEA (Recursive) and Batch EM (Batch) algorithms

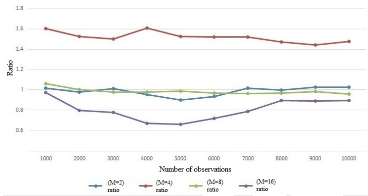

Figure 3: Ratio of SE of O_GHMM_PEA and Batch EM algorithms

state transition matrix. The results show that after processing 1000 and 10000 observations the state transition matrix is the same for both, the O_GHMM_PEA and batch EM algorithms.

Table 2: State transition probability matrix of the batch EM and O_GHMM_PEA algorithms

M = 2		
T=	Batch EM	O_GHMM_PEA
1000	$\begin{bmatrix} 0.497 & 0.503 \\ 0.472 & 0.528 \end{bmatrix}$	$\begin{bmatrix} 0.497 & 0.503 \\ 0.472 & 0.528 \end{bmatrix}$
10000	$\begin{bmatrix} 0.506 & 0.494 \\ 0.498 & 0.502 \end{bmatrix}$	$\begin{bmatrix} 0.506 & 0.494 \\ 0.498 & 0.502 \end{bmatrix}$

3.1.3 Experiments for state transition matrix calculation effectiveness

An experiment to demonstrate the effect of a state transition probability calculation using the Chapman-Kolmogorov equation was performed. O_GHMM_PEA was compared to the algorithm from [26] because it implements the classic forward-backward procedure skipping the backward part.

Several 3, 5, and 12 dimensional observation datasets were generated (T=800). All experiments were repeated one hundred times. Table 3 shows that the average standard error of O_GHMM_PEA algorithm is smaller than the algorithm from [26] for all three simulated datasets.

This experiment demonstrates the significance of the proposed state transition probability calculation when estimating HMM parameters in an online manner. It enhances the overall parameter estimation compared to an algorithm with only the forward procedure.

Table 3: Standard error of mean and covariance matrices

Algorithm	**Parameters**	$M = 3$	$M = 5$	$M = 12$
O_GHMM_PEA	μ	0.01	0.01	0.01
	σ	0.02	0.01	0.01
Stenger [26]	μ	0.20	0.17	0.72
	σ	0.28	0.10	0.11
Ratio	μ	0.04	0.04	0.01
	σ	0.06	0.12	0.04

3.1.4 Experiments for initial approximation

Experiments were performed to determine the effect on the O_GHMM_PEA convergence of the initial approximation dataset size. Multiple initial training datasets of various sizes (t = 50, 100, 200, 300, and 400) were used for calculations with two and five clusters of 2-dimensional and 8-dimensional data.

Figures 4 and 5 show that the standard error of the parameter estimates decreases as the total dataset size increases, even when the initial approximation dataset size is small. The parameter estimates converge to the original parameter values with every new observation. Nevertheless, when the number of clusters is small, the size of the training dataset can also be relatively small, even when the dimensions of the observation vectors increase.

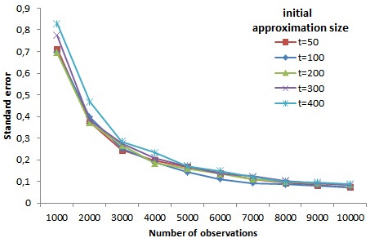

Figure 4: Initial approximation with 2-D data in 2 clusters

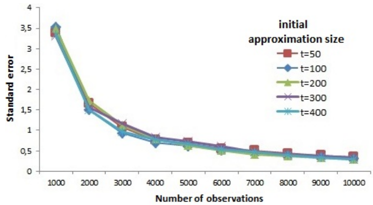

Figure 5: Initial approximation with 8-D data in 2 clusters

The experiments performed with the 2- and 8-dimensional data of 5 clusters show that the gap between the original and estimated parameter values decreases as the initial approximation size increases (see Fig. 6 and 7). If the initial training dataset is too small, the estimated parameters may converge to distorted local extremes of the likelihood function. Therefore, it must be of a size that would guarantee the algorithm's stability.

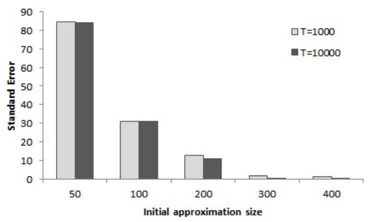

Figure 6: Initial approximation with 2-D data in 5 clusters

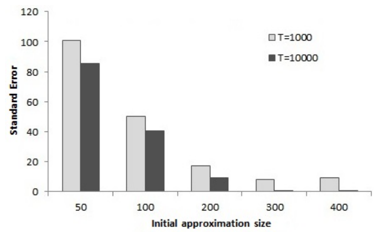

Figure 7: Initial approximation with 8-D data in 5 clusters

4 ONLINE DIRICHLET HMM PARAMETER ESTIMATION

4.1 Dirichlet distribution

Let $\mathbf{O} = (o_1, o_2, \ldots, o_m)$ be a random vector following a Dirichlet distribution $Dir(\alpha)$, where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_M)$. The joint density function is given by the following [31]: $p(o_1, o_2, \ldots, o_m) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{M} \Gamma(\alpha_i)} \prod_{i=1}^{M} o_i^{\alpha_i - 1}$, where $\alpha_0 = \sum_{i=1}^{m} \alpha_i$, $\alpha_i > 0, \forall i = 1 \ldots m$. $\sum_{i=1}^{m-1} o_i < 1$ and $o_m = 1 - \sum_{i=1}^{m-1} o_i$, where $0 < o_i < 1, \forall i = 1 \ldots m$.

Dirichlet distribution with the parameter vector $\alpha = (\alpha_1, \ldots, \alpha_M)$ can be represented as a distribution inside the simplex $D_m = \left\{(o_1, o_2, \ldots, o_m), \sum_{i=1}^{m-1} o_i < 1\right\}$ in $\mathbb{R}_+^m$. It means that this simplex limits the data between 0 and 1. Softmax normalization can be used to ensure that the sum of the vector elements will be 1 and all vector elements will be positive:

$$Softmax_i(\mathbf{O}) = \frac{e^{o_i}}{\sum_j e^{o_j}}. \tag{10}$$

4.2 Specification of the model

We present a novel online estimation algorithm for Dirichlet HMM parameters (O_DHMM_PEA). The HMM parameters are defined as continuous multivariate. In this case the Gaussian Mixture Densities output distribution is widely used. However, we assume that the distribution of the HMM output is Dirichlet.

We determine the following HMM parameters:

- N, which is the number of states in the HMM;
- the matrix of transition probabilities $\mathbf{A}$;
- the vector of the initial probability distribution π;

- and the Dirichlet distribution:

$$Dir(a) = \frac{\Gamma(\sum_{i=1}^{M} \alpha_i)}{\prod_{i=1}^{M} \Gamma(\alpha_i)} (1 - \sum_{i=1}^{M-1} o_i)^{\alpha_M - 1} \prod_{i=1}^{M-1} o_i^{\alpha_i - 1}, \quad (11)$$

where $\alpha_1, \ldots, \alpha_M, \alpha_i > 0$, $M > 2$, $o_1, \ldots, o_M, o_i \in (0, 1)$ and $\sum_{i=1}^{M} o_i = 1$.

Given a set of observed multivariate data vectors **O**, the parameters for a Dirichlet distribution (11) can be estimated by maximizing the log-likelihood function of the data:

$$\begin{aligned} logDir(\mathbf{O}|\alpha) = \ln\Gamma(\sum_{i=1}^{M} \alpha_i) - \sum_{i=1}^{M} \ln\Gamma(\alpha_i) + \\ + \left[(\alpha_M - 1)\ln(1 - \sum_{i=1}^{M-1} o_i)\right] + \sum_{i=1}^{M-1} \left[(\alpha_i - 1)\ln(o_i)\right]. \end{aligned} \quad (12)$$

Distribution parameter estimates have to maximize log-likelihood function:

$$logDir(\alpha) \to \max_{\alpha}. \quad (13)$$

The probability density of the observation is defined as follows: $L(\alpha, \pi) = -\ln\left[\sum_{q=1}^{N} \pi_q Dir(\mathbf{O}|\alpha_q)\right]$, where π_q is the probability of being in state q.

By calculating the derivative of the objective function with respect to α, we obtain the following:

$\frac{\partial}{\partial \alpha_j} logDir(O|\alpha) = \Psi\left(\sum_{i=1}^{M} \alpha_i\right) - \Psi(\alpha_j) + \ln(o_j), 1 \leq q \leq M$, $\Psi(.)$ is known as the digamma function.

Then $\frac{\partial}{\partial \alpha_j} L(\alpha, \pi) = \frac{\pi^{\langle q \rangle} Dir(O|\alpha_q) \frac{\partial}{\partial \alpha_j} logDir(O|\alpha)}{\sum_{q=1}^{N} \pi^{\langle q \rangle} Dir(O|\alpha_q)}, 1 \leq q \leq N$. We define the probability that the system is at state q in time step t given the observation sequence o. The model is $\frac{\pi_t^{\langle q \rangle} logDir(o_t|\alpha^{\langle q \rangle})}{\sum_{j=1}^{N} \pi_t^{\langle j \rangle} logDir(o_t|\alpha^{\langle j \rangle})}$, $1 \leq q \leq N$. Then the number of times the state trajectory is expected to transition from state q is $\sum_{t=1}^{T} \frac{\pi_t^{\langle q \rangle} logDir(o_t|\alpha^{\langle q \rangle})}{\sum_{j=1}^{N} \pi_t^{\langle j \rangle} logDir(o_t|\alpha^{\langle j \rangle})}, 1 \leq q \leq N$.

Since the full likelihood of each observation sequence is based on the summation of all possible state sequences, each observation is allocated to each state in proportion to the likelihood of the model being in that state at the time the variable was observed. Therefore, the summation of such weighted averages re-estimates the HMM's PDF parameters as follows:

$$\widehat{\alpha_i} = \Psi\Big(\sum_{m=1}^{M} \alpha_m^{\langle q\rangle}\Big) + \frac{\frac{1}{T}\sum_{t=1}^{T} \frac{\ln(o_t^{\langle s\rangle})\pi_t^{\langle q\rangle} logDir(o_t|\alpha^{\langle q\rangle})}{\sum_{j=1}^{N}\pi_t^{\langle j\rangle} logDir(o_t|\alpha^{\langle j\rangle})}}{\frac{1}{T}\sum_{t=1}^{T} \frac{\pi_t^{\langle q\rangle} logDir(o_t|\alpha^{\langle q\rangle})}{\sum_{j=1}^{N}\pi_t^{\langle j\rangle} logDir(o_t|\alpha^{\langle j\rangle})}}, \tag{14}$$

$1 \leq i \leq M-1$,

$$\widehat{\alpha_M} = \Psi\Big(\sum_{m=1}^{M} \alpha_m^{\langle q\rangle}\Big) + \frac{\frac{1}{T}\sum_{t=1}^{T} \frac{\ln(1-\sum_{s=1}^{M-1} o_t^{\langle s\rangle})\pi_t^{\langle q\rangle} logDir(o_t|\alpha^{\langle q\rangle})}{\sum_{j=1}^{N}\pi_t^{\langle j\rangle} logDir(o_t|\alpha^{\langle j\rangle})}}{\frac{1}{T}\sum_{t=1}^{T} \frac{\pi_t^{\langle q\rangle} logDir(o_t|\alpha^{\langle q\rangle})}{\sum_{j=1}^{N}\pi_t^{\langle j\rangle} logDir(o_t|\alpha^{\langle j\rangle})}}, \tag{15}$$

where q is state of the HMM, $1 \leq q \leq N$, $1 \leq s \leq M$, and $\pi_t^{\langle q\rangle}$ is the probability of being in state q in time frame t.

Note, that these formulas (14) and (15) contain $\pi_t^{\langle q\rangle}$, which represents the probability of being in state q at time t and calculated using the Chapman-Kolmogorov equation: $\pi_t = \mathbf{A} \cdot \pi_{t-1}$.

Formulas (14)-(15) are for a batch mode. We can use them to derive the recursive formulas for the HMM parameter estimation.

We assume that the parameter $\alpha_t^{\langle q,s\rangle}$ can be approximated using the previous parameter estimates $\alpha_{t-1}^{\langle q,s\rangle}$ and recursive equations for updating parameters:

$$\theta_t^{\langle q\rangle} = \pi_t^{\langle q\rangle} logDir\Big(o_t|\alpha^{\langle q\rangle}\Big), \tag{16}$$

$$\omega_t^{\langle q,s\rangle} = \omega_{t-1}^{\langle q,s\rangle} + \frac{1}{t}\Bigg(\frac{\ln(o_t^{\langle s\rangle})\theta_t^{\langle q\rangle}}{\sum_{j=1}^{N}\theta_t^{\langle j\rangle}} - \omega_{t-1}^{\langle q,s\rangle}\Bigg), 1 \leq q \leq N, \tag{17}$$

$$\omega_t^{\langle q,M\rangle} = \omega_{t-1}^{\langle q,M\rangle} + \frac{1}{t}\Bigg(\frac{\ln(1-\sum_{s=1}^{M-1} o_t^{\langle s\rangle})\theta_t^{\langle q\rangle}}{\sum_{j=1}^{N}\theta_t^{\langle j\rangle}} - \omega_{t-1}^{\langle q,M\rangle}\Bigg), \tag{18}$$

$$\gamma_t^{\langle q\rangle} = \gamma_{t-1}^{\langle q\rangle} + \frac{1}{t}\left(\frac{\theta_t^{\langle q\rangle}}{\sum_{j=1}^{N}\theta_t^{\langle j\rangle}} - \gamma_{t-1}^{\langle q\rangle}\right), \tag{19}$$

$$\alpha_t^{\langle q,s\rangle} = \Psi^{-1}\left[\Psi\left(\sum_{i=1}^{M}\alpha_i^{\langle q\rangle}\right) + \frac{\omega_t^{\langle q,s\rangle}}{\gamma_t^{\langle q\rangle}}\right], 1 \le s \le M, \tag{20}$$

$\Psi^{-1}(\cdot)$ – is the inverted digamma function.

Proof. It is easy to ascertain that recursive equations (17)-(19) can be obtained in the following way: $\gamma_t^{\langle q\rangle} = \gamma_{t-1}^{\langle q\rangle} + \frac{1}{t}\left(\frac{\theta_t^{\langle q\rangle}}{\sum_{j=1}^{N}\theta_t^{\langle j\rangle}} - \gamma_{t-1}^{\langle q\rangle}\right) = \frac{1}{t-1}\sum_{i=1}^{t-1}\frac{\theta_i^{\langle q\rangle}}{\sum_{j=1}^{N}\theta_i^{\langle j\rangle}} + \frac{1}{t}\left(\frac{\theta_t^{\langle q\rangle}}{\sum_{j=1}^{N}\theta_t^{\langle j\rangle}} - \frac{1}{t-1}\sum_{i=1}^{t-1}\frac{\theta_i^{\langle q\rangle}}{\sum_{j=1}^{N}\theta_i^{\langle j\rangle}}\right) = \frac{1}{t}\sum_{i=1}^{t}\frac{\theta_i^{\langle q\rangle}}{\sum_{j=1}^{N}\theta_i^{\langle j\rangle}}$. Equations (18)-(19) follow similarly.

The proposed online algorithm – Online Dirichlet Hidden Markov Model Parameter Estimation Algorithm (O_DHMM_PEA) – for the estimation of the Dirichlet HMM parameter consists of two main parts: model training and parameter re-estimation. Model training is an initial parameters estimation performed using a small fixed-size training dataset and formulas (16)-(20). The initial value of $\alpha_t^{\langle q,s\rangle}$ is obtained using the fixed point method. The initial estimates for the online algorithm are important in order to ensure its consistency and to prevent convergence to distorted local extremes of the likelihood function that are not the solution to the problem (13). After the initial estimates of the parameters are obtained, further classification and re-estimation of the parameters carries out by monitoring the process in real time.

The complexity of O_DHMM_PEA is linear relative to the number of observations. It requires only a fixed number of operations in every iteration where a new observation is processed.

4.3 Experimental results

4.3.1 Experiments to compare Dirichlet HMM to Gaussian HMM

To compare the modelling capabilities of the algorithms O_DHMM_PEA and O_GHMM_PEA, two datasets were created from different distributions. Two datasets A_Dir and B_Gaus each composed of 100 data blocks (each block contains T=1000 3-dimensional vectors) were generated following a Dirichlet and Gaussian distribution, respectively. These datasets were used to estimate the model parameters and perform observation recognition with the Dirichlet HMM and Gaussian HMM. In both cases, the initial approximation was set to 200 vectors. The average recognition rate was calculated after processing all data blocks. The recognition and estimation of the Dirichlet HMM parameters were performed after normalizing the observation vectors using the Softmax function (10) in order to make them fall into the interval of [0,1].

The recognition rate (see Table 4) of the A_Dir using the Dirichlet HMM is higher than the Gaussian HMM. Similarly, the recognition rate of the B_Gaus using the Gaussian HMM is higher than that of Dirichlet HMM. Therefore, we cannot adapt the Gaussian HMM to all practical problems, even though it is commonly used for modelling various practical areas.

Table 4: Average recognition rate of the experiments after one hundred experimental repetitions; 2-state HMM 3-dimensional data

		Model	
		Dirichlet PMM	***Gaussian PMM***
Dataset	***B_Gaus***	95.29%	99.53%
	A_Dir	84.40%	58.31%

4.3.2 Experiments to examine convergence property

To examine the convergence property of the O_DHMM_PEA algorithm, the recognition rate was calculated and the parameter estimates were compared to original model parameters. During the experiment, the initial approximation was performed using 1000 observations. After the initial approximation, the re-estimation was performed and the recognition rate was calculated after each 1000 observations were processed.

Three-dimensional observations were generated from a 3-state HMM with the following parameters: $\mathbf{A} = \begin{bmatrix} 0.5 & 0.4 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.1 & 0.7 \end{bmatrix}$, $\pi = \begin{bmatrix} 0.4 \\ 0.3 \\ 0.3 \end{bmatrix}$, $\alpha^1 = \begin{bmatrix} 8 \\ 2 \\ 2 \end{bmatrix}$, $\alpha^2 = \begin{bmatrix} 80 \\ 80 \\ 80 \end{bmatrix}$, $\alpha^3 = \begin{bmatrix} 2 \\ 8 \\ 8 \end{bmatrix}$.

Three-dimensional observations were generated from a 5-state model with the following parameters:

$$\mathbf{A} = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.1 & 0.1 \\ 0.1 & 0.4 & 0.2 & 0.2 & 0.1 \\ 0.1 & 0.3 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.1 & 0.3 & 0.3 & 0.1 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \end{bmatrix}, \pi = \begin{bmatrix} 0.4 \\ 0.2 \\ 0.2 \\ 0.1 \\ 0.1 \end{bmatrix}, \alpha^1 = \begin{bmatrix} 8 \\ 2 \\ 2 \end{bmatrix}, \alpha^2 = \begin{bmatrix} 20 \\ 20 \\ 20 \end{bmatrix},$$

$\alpha^3 = \begin{bmatrix} 2 \\ 8 \\ 8 \end{bmatrix}$, $\alpha^4 = \begin{bmatrix} 60 \\ 60 \\ 60 \end{bmatrix}$, ir $\alpha^5 = \begin{bmatrix} 5 \\ 15 \\ 5 \end{bmatrix}$.

Recognition accuracy was calculated with confidence intervals as depicted bellow:

$$AT \pm 1.96 \cdot \sqrt{\frac{AT \cdot (1 - AT)}{T}} \tag{21}$$

4.3.2.1 Experimental results

The results in Table 5 and 6 show the recognition rate of O_DHMM_PEA. The results in Figure 8 show that the O_DHMM_PEA algorithm converges to the real parameter values. The standard error of the parameter estimates decreases as T increases.

The results of the 5-state HMM 3-dimensional observations and the 3-state HMM 3-dimensional observations show that number of states is an important factor in the convergence and efficiency of the algorithm. The fact that the algorithm converges to the original values of the parameter is seen from the declining standard error of calculated values (Fig. 9) and the rate of recognition, which is gradually increasing. The experiments with the synthetic data clearly demonstrate O_DHMM_PEA efficiency.

Table 5: Recognition rate (3-state 3-D data)

T=	Accuracy
1000	$98\% \pm 0.008$
2000	$98\% \pm 0.006$
3000	$98.1\% \pm 0.004$
4000	$98.1\% \pm 0.004$

Table 6: Recognition rate (5-state 3-D data)

T=	Accuracy
1000	$77.2\% \pm 0.02$
2000	$77.5\% \pm 0.018$
5000	$77.8\% \pm 0.011$
9000	$77.8\% \pm 0.008$

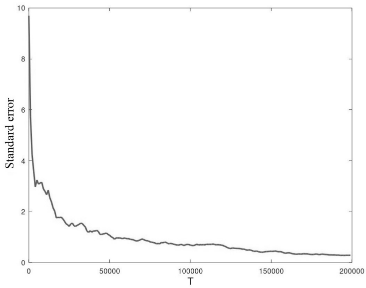

Figure 8: SE (3-state 3-D data)

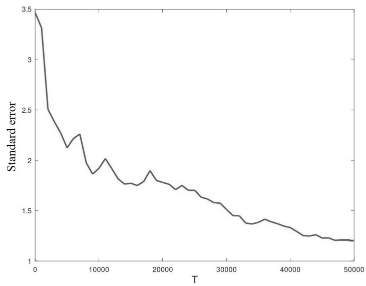

Figure 9: SE (5-state 3-D data)

5 APPLICATIONS

5.1 Isolated word recognition

Automatic speech recognition (ASR) is a pattern recognition task which aims to classify input data into classes based on certain features. ASR receives input of a speech signal, processes it and outputs the text corresponding to the input. Traditional methods of speech learning such as deep neural networks, linear-time-scaled word-template matching and HMM require static training dataset to effectively learn parameters of the speech model. These methods have at least the quadratic complexity, as the number of calculations that is needed for every learning iteration depends on the dataset size.

Online learning could be the solution in cases when the number of speech samples for the training is too small to be practically used in recognition systems. This could provide practical collection of speech data in real time without retraining the speech model.

In this work, isolated word recognition (IWR), which is an ASR subset was performed using the O_GHMM_PEA algorithm for the estimation of HMM parameters. In the IWR system the input data are viewed as independently processed words, and previously uttered words do not affect the recognition.

The input data is a raw speech file that is translated into a acoustic feature vector and processed over time. The input data is a raw speech file that is converted into an acoustic feature vector and is processed over time. A left-to-right discrete-state HMM is used to model each word. In this case $\mathbf{A}_{i,j} = 0, j < i$. The sum of each matrix row elements is equal to one.

5.1.1 O_GHMM_PEA application to IWR

ASR typically consists of multiple phases – primary processing, analysing the signal and final processing. Primary processing involves pre-processing the signal (such as noise reduction), signal analysis consists of extracting features from the speech signal, and final processing consists of a speech recognition engine with an acoustic model, language model and grammar in it. If IWR is created then a language model and grammar are not needed. IWR systems recognize single words separated by silence. We will only address the acoustic model (leaving out the language model and grammar) since the O_GHMM_PEA algorithm is being applied to IWR.

In order to apply the O_GHMM_PEA algorithm to IWR, we have to consider the processing of data. Isolated words can be processed in two ways – at the level of the symbol/phoneme or in words blocks. To adapt the O_GHMM_PEA algorithm for IWR, the data will be processed in blocks/words. The first part of the O_GHMM_PEA algorithm performs an initial approximation to the HMM parameters using training data. In the second part of the algorithm a recognition procedure was implemented to identify observations. The identification can be performed with the Viterbi algorithm [32]. The HMM parameters are then updated according to the identified word. The scheme for the adapted O_GHMM_PEA algorithm is presented in Figure 10.

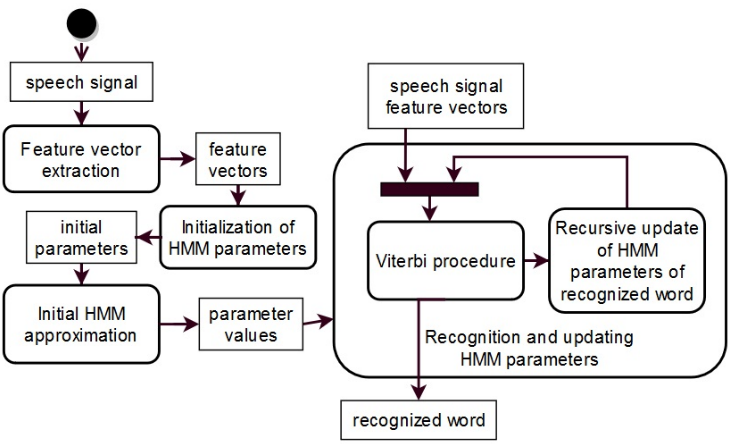

Figure 10: O_GHMM_PEA adapted to isolated word recognition.

5.1.2 IWR experiments

5.1.2.1 Experimental setup

IWR experiments were modelled in Matlab. The prototype for isolated word recognition was implemented. All IWR experiments were conducted applying the following data. The initial state distribution vector was set to $\pi = \begin{bmatrix} 0 \, 0.8 \, 0.2 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \, 0 \end{bmatrix}^T$. The state transition probability matrix was set to

$$\mathbf{A} = \begin{bmatrix} 0 & 0.8 & 0.2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6 & 0.3 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.6 & 0.3 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.6 & 0.3 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.6 & 0.3 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.6 & 0.3 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0.3 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.6 & 0.3 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.67 & 0.33 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The value of the stopping criterion was set to $\epsilon = 0.01$.

Word recognition rate is computed as: $WRR = \frac{N-S-D-I}{N}$, where S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, N is the number of words in the reference ($N = S + D + C$).

5.1.2.2 TIDIGITS dataset

The O_GHMM_PEA algorithm was adapted to perform recognition and parameter estimation for isolated speech data. Training and testing were performed with a subset of the TIDIGITS dataset [33]. It consists of 208 speakers (94 men, 114 women) each pronouncing 22 digit sequences (from zero to nine).

The experiments were conducted in the following manner. Firstly, fixed initial training datasets of various sizes ($100 \leq t \leq 2000$ words) were chosen to perform the calculations. Secondly, further training and recognition were performed with a 1500 word dataset. The word recognition rate (WRR, recognition accuracy) was calculated during the second part of the algorithm. The feature vectors were extracted of 39 features in the MFCC format. The main focus of the experiment was to explore the influence of the initial training dataset size on the word recognition rate. The results are presented in Table 7.

Table 7: IWR with the implemented O_GHMM_PEA algorithm

Size (in words) of the initial training dataset	**Recognition rate (%)**	**Confidence interval**
100	92.53	±0.051
500	94.33	±0.020
1000	95.87	±0.012
1500	97.60	±0.007
2000	97.27	±0.007

For an initial dataset size of 100 words, the word recognition rate was 92.53%. Increasing the initial dataset size to 2000 words, WRR increased to 97%. When the initial dataset size increases from 100 to 1500 words, WRR is continuously increasing as well. Thus, it is important to choose an appropriate size for the initial parameter estimation dataset as it prevents the algorithm from converging to a distorted local extreme of the objective function.

5.1.2.3 Spoken Arabic Digits dataset

Additional experiments were performed with the Spoken Arabic Digits dataset [34]. The training dataset consists of 8143 observations, which were used for initial HMM parameter learning. The testing dataset consists of 2665 observations, which were used for re-estimation and recognition in real-time.

For modelling, we used a multivariate Gaussian HMM with multivariate parameters. The dataset consists of a feature vector with 12 features. Thus, HMM states are modelled with a 12-dimensional mean vector and covariance matrix. Each word (digit) was modelled as a 10-state HMM.

The results showed that the recognition rate of isolated words recognition was 91.86% with confidence interval of ± 0.011. Out of 2200 words, the O_GHMM_PEA algorithm classified correctly 2021 words.

5.2 Occupancy detection

In recent years, commercial and residential occupancy detection methods have become widespread. IIt helps the automation programs to control the energy consumption of the building. The available occupancy data can also help to control the use of building energy, temperature (thermostats), HVAC systems (heating - ventilation - air conditioning), lighting and other devices. Currently, there are automated systems with integrated occupancy detection to effectively control room temperature, air conditioning, and lighting.

Various machine learning algorithms are used in the literature, such as support vector machines (SVM), artificial neural networks (ANN), decision trees (DT), agent-based models, etc. For example, the article [35] uses Conditional Random Field model and the Hidden Markov Support Vector Machine (HMSVM) to estimate the number of users in a three-person residence using PIR motion sensor data that are accessible through the alarm system. However, there still remains the problem of limited amount of data for training machine learning algorithms. The article [36] proposed an employment estimation system based on energy consumption data in order to solve the problems of learning from a limited amount of data.

Gradual application of HMM parameter estimation algorithms could solve the problem of limited training data. In the case of a incremental algorithm, the data already collected from the sensors would not be stored. An online algorithm could process real-time data from various sensors and simultaneously determine room occupancy.

This work deals with the occupancy detection task. This task includes the binary classification: the room is occupied, the room is unoccupied. Machine learning algorithms are trained with a data extracted from smart-meters (temperature, relative humidity, lighting, carbon dioxide concentration, water vapor content in the air).

5.2.1 Experimental setup

The effectiveness of the O_GHMM_PEA and O_DHMM_PEA algorithms for occupancy detection is analysed. The features in the dataset that are most useful in training of the algorithm are analysed. O_GHMM_PEA and O_DHMM_PEA are compared to other machine learning methods (SVM, LR, ANN). Experiments were performed with occupancy detection dataset [37]. The data consists of two classes that describe the occupancy of the room: occupied and unoccupied. If we consider the occupancy detection as a process when space becomes occupied or unoccupied over time, then we will be able to model it with two-state HMM. Dataset features are:

#1 Date (year-month-day);
#2 Time (hour:minute:second);
#3 Temperature, in Celsius;
#4 Relative Humidity (%);
#5 Light, in Lux;
#6 CO2, in ppm;
#7 Humidity Ratio, in kg-water-vapour/kg-air.

The training dataset consists of 8143 observations. The testing dataset consists of 2665 observations, which were used for the real-time re-estimation and recognition.

Data was normalized for the O_DHMM_PEA algorithm using Softmax (10). In case of O_GHMM_PEA, data normalization was not applied. Model parameter values were initialized using the method of moments and training dataset. The stopping criterion for both algorithms was set to $\epsilon = 0.01$. Classification was performed using Bayes rule; the algorithm efficiency metrics were calculated and feature selection experiments were conducted.

The standard efficiency estimation methodology of the classification algorithm is used – the confusion matrix (see Table 8) and the main indicators, which are calculated from the values of the said matrix – accuracy, precision, recall, and F-score:

$$Precision = \frac{TP}{TP + FP}, \tag{22}$$

$$Recall = \frac{TP}{TP + FN}, \tag{23}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{24}$$

$$Fscore = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{25}$$

Table 8: Confusion matrix for occupancy detection

		Classified	
		Unoccupied	**Occupied**
Actual	**Unoccupied**	True positive (TP)	False negative (FN)
	Occupied	False positive (FP)	True negative (TN)

5.2.1.1 Experimental results

The first experiment was performed to select data features for classification. The correlation coefficients and P-values between the occupancy detection dataset features were calculated. Table 9 shows that the correlation coefficient values are normal in most cases. There is a very strong correlation (0.96) between features #4 and #7, and a strong correlation between the two feature pairs: #1 and #4, and #1 and #7.

P-values of correlation for these features were calculated and they were lower than the standard significance level (0.05). Thus, the corresponding correlations between these features are significant.

Table 9: Occupancy detection feature correlation coeficient matrix

Feature	#1	#2	#3	#4	#5	#6
#2	-0.01					
#3	-0.08	0.26				
#4	0.74	0.02	-0.14			
#5	-0.05	0.09	0.65	0.04		
#6	0.20	0.21	0.56	0.44	0.66	
#7	0.70	0.1	0.15	0.96	0.23	0.63

Based on the fact that there is a strong correlation between these features, we can say that removing the corresponding feature from the pair of strongly correlated features would not change the recognition accuracy after training the algorithm or, in best-case scenario, increase it. Further experiments were performed based on these results. Model training and re-estimation were performed by removing these features from the feature vector one by one, respectively. The efficiency metrics are presented in Table 10.

Table 10: Occupancy detection results of O_DHMM_PEA with the modified feature set.

	All	Without #7	Without #4	Without #1
Accuracy	0.98	0.97	0.97	0.98
Precision	0.94	0.93	0.93	0.94
Recall	1	1	1	1
Fscore	0.97	0.96	0.96	0.97

The O_DHMM_PEA and O_GHMM_PEA algorithms were compared. Algorithm efficiency metrics were calculated and are listed in Table 11. O_DHMM_PEA gives better performance than

O_GHMM_PEA in all cases. It shows that the O_DHMM_PEA algorithm is more accurate than the O_GHMM_PEA algorithm.

No online type algorithms have been found in the literature to solve the occupancy detection problem. Thus, these results were compared with the results in the article [38]. Table 12 columns lists best F-score results obtained with a linear regression classifier (LR), support vector classifier (SVM), artificial neural networks (ANN) [38]. We can see that the O_DHMM_PEA algorithm is not inferior to the already existing algorithms described in the literature and is effective in solving the problem of occupancy detection.

Table 11: Efficiency metrics of O_DHMM_PEA and O_GHMM_PEA for occupancy detection.

	Dir	**Gaus**
Accuracy	0.98	0.87
Precision	0.94	0.83
Recall	1	0.81
F-score	0.97	0.82

Table 12: Highest F-Score of algorithms (LR, SVM, and ANN) in [38] for occupancy detection.

	LR	**SVM**	**ANN**
Light	95.6	95.55	95.32
Temp	89.8	89.71	89.72

5.3 Pulsar identification

Pulsars are a rare type of a neutron star that produce radio emission detectable here on Earth [39, 40]. Searching for pulsars is not an easy task. Pulsar discovery involves the identification of periodic signals in monitoring data. This data is then reduced to a set of diagnostic values and graphs called a candidate [41]. Unfortunately, most candidates are caused by radio frequency interference (RFI) and noise, which are not real pulsars. Human experts can select the pulsar candidates but it is a subjective and time-consuming error-prone process [40, 41]. In this work, online algorithms (O_DHMM_PEA and O_GHMM_PEA) are applied to pulsar identification.

5.3.1 Experimental setup

The effectiveness of the O_GHMM_PEA and O_DHMM_PEA algorithms for pulsar identification is analysed. The features in the data set that are most useful in training of the algorithm are analysed. The proposed algorithms are tested with the HTRU2 dataset [40] and compared to other classifiers described in the literature.

HTRU2 dataset contains 16259 spurious examples caused by RFI/noise, and 1639 real pulsar examples. Each candidate is described by 8 continuous variables:

#1 Mean of the integrated profile.

#2 Standard deviation of the integrated profile.

#3 Excess kurtosis of the integrated profile.

#4 Skewness of the integrated profile.

#5 Mean of the DM-SNR curve.

#6 Standard deviation of the DM-SNR curve.

#7 Excess kurtosis of the DM-SNR curve.

#8 Skewness of the DM-SNR curve.

The purpose of the classification is to identify given candidates as pulsar and non-pulsar. The dataset was divided into training (T=7898) and testing (T=10000) subsets. Data was normalized for O_DHMM_PEA using Softmax (10). In case of O_GHMM_PEA, data normalization was not applied. Model parameter values were initialized using the method of moments and training dataset. The stopping criterion for both algorithms was set to $\epsilon = 0.01$. Classification was performed using the Bayes rule. Then the algorithm efficiency metrics were calculated and feature selection experiments were conducted.

The standard efficiency metrics of the classification algorithm is used – the confusion matrix (see Table 13) and the main indicators, which are calculated from the values of the said matrix: accuracy, precision, recall, F-score, specificity, false positive rate (FPR) and G-mean.

Table 13: Confusion matrix for pulsar identification

		Classified	
		Pulsar	**Non-pulsar**
Actual	**Pulsar**	True positive (TP)	False negative (FN)
	Non-pulsar	False positive (FP)	True negative (TN)

$$FPR = \frac{FP}{TN + FP}, \tag{26}$$

$$Specificity = \frac{TN}{FP + TN}, \tag{27}$$

$$GMean = \sqrt{\left(\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}\right)}. \tag{28}$$

5.3.1.1 Experimental results

O_DHMM_PEA and O_GHMM_PEA were implemented for pulsar identification experiments. The experimental results are listed in Table 14. F-score of O_DHMM_PEA and O_GHMM_PEA is not high. However FPR of O_DHMM_PEA is significantly low – 0.07. The recognition accuracy of O_DHMM_PEA is 10% higher than O_GHMM_PEA.

No online type algorithms have been found in the literature to solve the pulsar identification task. Thus, O_DHMM_PEA was compared to the Fuzzy KNN algorithm [42] (in this article experiments were performed with HTRU2 dataset). Results are listed in Table 15.

An experiment for feature selection of HTRU2 dataset was performed. Table 16 shows that the correlation coefficient values are normal in most cases. There is a very strong correlation between the pairs of features #3 and #4, #8 and #7. P-values of correlation for these pairs of features were calculated and they were lower than the standard significance level (0.05). Thus, the corresponding correlations between these features are significant.

Table 14: Efficiency metrics of O_DHMM_PEA and O_GHMM_PEA for pulsar identification.

	O_DHMM_PEA	**O_GHMM_PEA**
Accuracy	0.92	0.82
FPR	0.07	0.18
Precision	0.43	0.23
Recall	0.84	0.91
F-score	0.57	0.37
Specificity	0.93	0.82
G-mean	0.89	0.86

Table 15: O_DHMM_PEA and Fuzzy KNN [42] results for HTRU2.

	Dirichlet PMM	**Fuzzy KNN**
Accuracy	0.93	0.98
F-score	0.57	0.87
Gmean	0.88	0.96
FPR	0.07	0.17

Table 16: HTRU2 feature correlation coeficient matrix

Feature	**#1**	**#2**	**#3**	**#4**	**#5**	**#6**	**#7**
#2	0.54						
#3	-0.87	-0.52					
#4	-0.73	-0.54	0.94				
#5	-0.29	0.00	0.41	0.41			
#6	-0.30	-0.04	0.43	0.41	0.79		
#7	0.23	0.02	-0.34	-0.32	-0.61	-0.81	
#8	0.14	0.02	-0.21	-0.20	-0.35	-0.57	0.92

Further experiments were performed after removing #4 and #8 features from dataset. The results are listed in Table 17. It shows that the removal of #8 feature increases the accuracy. The removal of the #8 and #4 features together does not change the recognition accuracy. The removal of the #8 feature slightly improves the precision, but decreases the G-mean compared to the unmodified feature set.

Experiments with a modified dataset showed that by removing attribute #8, we can slightly improve the recognition accuracy. Also, the removal of this feature reduces the amount of computer resources used for calculations, as calculations are performed with smaller matrices and vectors.

Table 17: Pulsar identification results of O_DHMM_PEA with a modified feature set.

	All	**Without #4**	**Without #8**	**Without #4 & #8**
Accuracy	0.92	0.90	0.93	0.92
FPR	0.06	0.09	0.06	0.06
Precision	0.43	0.35	0.44	0.39
Recall	0.82	0.85	0.82	0.73
Fscore	0.57	0.50	0.57	0.51
Specificity	0.93	0.90	0.93	0.93
Gmean	0.88	0.88	0.87	0.82

6 CONCLUSIONS

The dissertation examines the problem of online estimation of hidden Markov model parameters. Algorithms for online estimation of hidden Markov model parameters with multivariate Gaussian and non-Gaussian (Dirichlet) output probability distributions have been created.

The main results obtained are:

- Based on the maximum likelihood method and the batch EM algorithm, the O_GHMM_PEA and O_DHMM_PEA algorithms for estimating HMM parameters were developed, where the state output distributions are multivariate Gaussian and Dirichlet, respectively.
- The state transition probabilities of HMM are proposed to be calculated based on the Chapman-Kolmogorov equation. It replaces the traditional forward-backward procedure commonly used in Baum-Welch algorithms and the forward procedure in online algorithms.
- An experimental study of the proposed online algorithms was performed with synthetic data by the Monte-Carlo method, datasets of isolated words recognition, occupancy detection and pulsar identification.

The following conclusions were formulated with experimental results confirming them:

- The online HMM parameters estimation algorithm with Gaussian output is faster than the batch EM algorithm and has equivalent classification accuracy.
  - Computational time of the online algorithm (O_GHMM_PEA) is higher (3–9 times) than that of the batch EM algorithm when observations of a different number of dimensions are processed.
  - The state transition probability matrices of batch and online (O_GHMM_PEA) algorithms obtained during the

    estimation of HMM parameters by processing up to 10,000 observations do not differ.
  - The ratio of standard errors of model parameter estimates obtained by online (O_DHMM_PEA) and batch EM algorithms indicates that the online algorithm gives parameter estimates that differ minimally from the batch algorithm. The ratio of standard errors of online and batch algorithms ranges from 1.1 to 1.5 times, depending on the dimensions and standard deviation of the observations.
  - The results of the comparison of the O_GHMM_PEA with Stenger algorithm showed that the calculation of the state transition probabilities with the Chapman-Kolmogorov equation improves the approximation of the parameters to the true values of the model parameters compared to the algorithm where the state transition probability matrix is calculated with the forward procedure. After processing three-dimensional observations with the O_GHMM_PEA algorithm, the standard error of HMM parameter estimates was 0.008, and that of the Stenger algorithm - 0.19.
- There exists a sufficient initial approximation dataset of the algorithm that ensures the stability of the algorithm and prevents it from converging to the distorted local extremes of the likelihood function.
  - The standard error of the parameter estimates decreases when the total dataset size is increasing if the minimum initial approximation dataset size is sufficient. In case of processing the five-cluster two-dimensional and eight-dimensional observations, the standard error of the model parameter estimates does not differ, while the size of the initial dataset is 300 observations, and it is increased to 400.
  - With a small number of clusters (HMM states), the size of the dataset used for initial training can also be small,

regardless of the number of dimensions of the observation vectors. However, increasing the number of clusters is to increase the initial approximation dataset.

  - Applying the O_GHMM_PEA to isolated word recognition showed that the recognition accuracy was 97% for the TI-DIGITS and 91% for the Spoken Arabic Digits dataset.

- Standard deviation and the number of dimensions determine the efficiency of online HMM parameter estimation – as the number of dimensions increases, the standard error of HMM parameter estimates increases too.
  - The standard error, when the observations are two-dimensional and the standard deviation is 20, is about 4-6 times larger than the standard error obtained after processing 16-dimensional data with a standard deviation of 50.
  - As the standard deviation and dimensions of data increase, the computational time of the algorithm increases accordingly.
  - As the amount of data processed increases, the standard error of the model parameter estimates decreases accordingly.

Experimental studies with the online Dirichlet HMM parameter estimation algorithm were performed and the following conclusions were formulated:

- Comparison of the online Dirichet HMM parameter estimation algorithm and the online Gaussian HMM parameter estimation algorithm showed that widespread Gaussian HMM cannot be applied to problems where the observations are of Dirichlet distribution; in this case Dirichlet HMM is more efficient with 26% higher classification accuracy.
- The recognition accuracy does not decrease and the standard error of parameter estimates decreases as the training data

increases. The standard error of the HMM parameter estimates reduced 2.5 times after processing 4000 observations during the parameter re-estimation.

- The recognition accuracy of Dirichlet HMM was 97% in the case of the occupancy detection, and 93% in the case of the pulsar identification.
- Modifying the feature set affects the classification accuracy. In case of occupancy detection, the classification accuracy does not change by removing the features ”date”, ”relative humidity” and ”humidity ratio” from the feature set. In the case of pulsar identification, modifying the feature set (removing the feature ”skewness of the DM-SNR curve”) slightly (0.01%) improves the recognition accuracy.

Research of the online algorithms is useful in the development and application of online equivalents of other types of algorithms for real-time classification.

REFERENCES

[1] O. Cappé, “Online EM Algorithm for Hidden Markov Models,” *Journal of Computational and Graphical Statistics*, vol. 20, no. 3, pp. 728–749, 2011.

[2] O. Cappé, E. Moulines, and T. Rydén, *Inference in hidden Markov models*. Springer, 2010.

[3] V. Krishnamurthy and J. B. Moore, “On-line Estimation of Hidden Markov Model Parameters Based on the Kullback-Leibler Information Measure,” *IEEE Trans. Signal Processing*, vol. 41, pp. 2557–2573, 1993.

[4] T. Ryden, “On Recursive Estimation for Hidden Markov Models,” *Stochastic Processes and their Applications*, vol. 66, no. 1, pp. 79 – 96, 1997.

[5] G. Mongillo and S. Deneve, “Online Learning with Hidden Markov Models,” *Neural Computation*, vol. 20, no. 7, pp. 1706–1716, 2008.

[6] V. B. Tadic, “Analyticity, Convergence, and Convergence Rate of Recursive Maximum-Likelihood Estimation in Hidden Markov Models,” *IEEE Transactions on Information Theory*, vol. 56, no. 12, pp. 6406–6432, 2010.

[7] F. LeGland and L. Mevel, “Recursive Estimation in Hidden Markov Models,” in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 4, pp. 3468–3473 vol.4, Dec 1997.

[8] I. Collings, V. Krishnamurthy, and J. B. Moore, “On-line Identification of Hidden Markov Models via Recursive Prediction Error Techniques,” *Signal Processing, IEEE Transactions on*, vol. 42, pp. 3535–3539, 1995.

[9] J. J. Ford and J. B. Moore, “Adaptive Estimation of HMM Transition Probabilities,” *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1374–1385, 1998.

[10] J. Lai, J. J. Ford, P. O’Shea, and L. Mejias, “Vision-based Estimation of Airborne Target Pseudobearing Rate Using Hidden Markov Model Filters,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2129–2145, 2013.

[11] A. Kontorovich, B. Nadler, and R. Weiss, “On Learning Parametric-Output HMMs,” in *Proceedings of the 30th International Conference on Machine Learning* (S. Dasgupta and D. McAllester, eds.), vol. 28 of *Proceedings of Machine Learning Research*, (Atlanta, Georgia, USA), pp. 702–710, PMLR, 17–19 Jun 2013.

[12] W. Zucchini, I. L. Macdonald, and R. Langrock, *Hidden Markov Models for Time Series: An Introduction Using R*. Crc Press, 2016.

[13] N. M. Nasrabadi, "Pattern Recognition and Machine Learning," *Journal of Electronic Imaging*, vol. 16, no. 4, 2007.

[14] R. Mattila, C. R. Rojas, V. Krishnamurthy, and B. Wahlberg, "Identification of Hidden Markov Models Using Spectral Learning with Likelihood Maximization," *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, Dec 2017.

[15] T. Cheng, S. Dixon, and M. Mauch, "Improving Piano Note Tracking by HMM Smoothing," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 2009–2013, Aug 2015.

[16] L. R. L. Rodrigues and E. L. Pinto, "HMM Models and Estimation Algorithms for Real-time Predictive Spectrum Sensing and Cognitive Usage," in *XXXV SIMPOSIO BRASILEIRO DE TELECOMUNICACOES E PROCESSAMENTO DE SINAIS - SBrT2017*, 2017.

[17] M. Stamp, "A Revealing Introduction to Hidden Markov Models," 2015.

[18] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[19] W. Khreich, E. Granger, A. Miri, and R. Sabourin, "A Survey of Techniques for Incremental Learning of HMM Parameters," *Information Sciences*, vol. 197, p. 105–130, Aug 2012.

[20] Y. Ephraim and N. Merhav, "Hidden Markov Processes," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1518–1569, 2002.

[21] Y.-A. Ma, N. J. Foti, and E. B. Fox, "Stochastic Gradient MCMC Methods for Hidden Markov Models," in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 2265–2274, PMLR, 06–11 Aug 2017.

[22] R. Mattila, C. R. Rojas, and B. Wahlberg, "Evaluation of Spectral Learning for the Identification of Hidden Markov Models," *IFAC-PapersOnLine*, vol. 48, no. 28, pp. 897 – 902, 2015. 17th IFAC Symposium on System Identification SYSID 2015.

[23] Y. C. Subakan, J. Traa, P. Smaragdis, and D. Hsu, "Method of Moments Learning for Left-to-right Hidden Markov Models," in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5, Oct 2015.

[24] Qiang Huo and Chin-Hui Lee, "On-line Adaptive Learning of the Continuous Density Hidden Markov Model Based on Approximate Recursive Bayes Estim-

ate," *IEEE Transactions on Speech and Audio Processing*, vol. 5, pp. 161–172, March 1997.

[25] A. Bietti, F. Bach, and A. Cont, "An Online EM Algorithm in Hidden (semi-)Markov Models for Audio Segmentation and Clustering," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1881–1885, April 2015.

[26] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. M. Buhmann, "Topology Free Hidden Markov Models: Application to Background Modeling," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 1, no. C, pp. 294–301, 2001.

[27] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*. Springer, 2011.

[28] R. M. Neal and G. E. Hinton, *A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants*, pp. 355–368. Springer Netherlands, 2012.

[29] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2009.

[30] O. Leveque, "Lectur Notes on Markov Chains," 2011.

[31] B. A. Frigyik, A. Kapila, and M. R. Gupta, "Introduction to the Dirichlet Distribution and Related Processes," *UWEE Technical Report*, 2010.

[32] R. E. Gruhn, W. Minker, and S. Nakamura, *Automatic Speech Recognition*, pp. 5–17. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[33] R. G. Leonard and G. Doddington, "[dataset] TIDIGITS LDC93S10," 1993.

[34] D. Dua and C. Graff, "[dataset] Spoken Arabic Digits in UCI Machine Learning Repository," 2017.

[35] L. Yang, K. Ting, and M. B. Srivastava, "Inferring Occupancy from Opportunistically Available Sensor Data," in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 60–68, IEEE, 2014.

[36] M. Jin, R. Jia, and C. J. Spanos, "Virtual Occupancy Sensing: Using Smart Meters to Indicate Your Presence," *IEEE Transactions on Mobile Computing*, vol. 16, pp. 3264–3277, Nov 2017.

[37] L. Candanedo Ibarra and V. Feldheim, "Accurate Occupancy Detection of an Office Room from Light, Temperature, Humidity and CO2 Measurements using Statistical Learning Models," *Energy and Buildings*, vol. 112, 2015.

[38] B. Abade, D. Perez Abreu, and M. Curado, "A non-Intrusive Approach for Indoor Occupancy Detection in Smart Environments," *Sensors*, vol. 18, no. 11, p. 3953, 2018.

[39] A. Cameron, D. Champion, M. Kramer, M. Bailes, E. Barr, C. Bassa, S. Bhandari, N. Bhat, M. Burgay, S. Burke-Spolaor, *et al.*, "The High Time Resolution Universe Pulsar Survey–XIII. psr j1757- 1854, the Most Accelerated Binary Pulsar," *Monthly Notices of the Royal Astronomical Society: Letters*, vol. 475, no. 1, pp. L57–L61, 2018.

[40] R. Lyon, B. Stappers, S. Cooper, J. Brooke, and J. Knowles, "Fifty Years of Pulsar Candidate Selection: From Simple Filters to a New Principled Real-time Classification Approach," *Monthly Notices of the Royal Astronomical Society*, vol. 459, p. stw656, 04 2016.

[41] V. Morello, E. D. Barr, M. Bailes, C. M. Flynn, E. F. Keane, and W. van Straten, "SPINN: a Straightforward Machine Learning Solution to the Pulsar Candidate Selection Problem," *Mon. Not. Roy. Astron. Soc.*, vol. 443, no. 2, pp. 1651–1662, 2014.

[42] T. M. Mohamed, "Pulsar Selection Using Fuzzy KNN Classifier," *Future Computing and Informatics Journal*, vol. 3, no. 1, pp. 1 – 6, 2018.

[43] G. K. D. Teyou, "Deep Learning Acceleration Techniques for Real-time Mobile Vision Applications," 2019.

[44] R. Nishihara, P. Moritz, S. Wang, A. Tumanov, W. Paul, J. Schleier-Smith, R. Liaw, M. Niknami, M. I. Jordan, and I. Stoica, "Real-time Machine Learning: The Missing Pieces," in *Proceedings of the 16th Workshop on Hot Topics in Operating Systems*, HotOS '17, (New York, NY, USA), p. 106–110, Association for Computing Machinery, 2017.

[45] L. Xie, V. A. Ugrinovskii, and I. R. Petersen, "Finite Horizon Robust State Estimation for Uncertain Finite-alphabet Hidden Markov Models with Conditional Relative Entropy Constraints," *SIAM Journal on Control and Optimization*, vol. 47, no. 1, pp. 476–508, 2008.

[46] J. J. Ford, V. Ugrinovskii, and J. Lai, "An Infinite-Horizon Robust Filter for Uncertain Hidden Markov Models with Conditional Relative Entropy Constraints," in *2011 Australian Control Conference*, pp. 499–506, IEEE, 2011.

[47] W. L. Kendall, G. C. White, J. E. Hines, C. A. Langtimm, and J. Yoshizaki, "Estimating Parameters of Hidden Markov Models Based on Marked Individuals: Use of Robust Design Data," *Ecology*, vol. 93, no. 4, pp. 913–920, 2012.

[48] J. Unnikrishnan, V. V. Veeravalli, and S. P. Meyn, "Minimax Robust Quickest Change Detection," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1604–1614, 2011.

[49] J. Dong, M. Verhaegen, and F. Gustafsson, "Robust Fault Detection with Statistical Uncertainty in Identified Parameters," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5064–5076, 2012.

[50] B. C. Levy, "Robust Hypothesis Testing with a Relative Entropy Tolerance," *IEEE Transactions on Information Theory*, vol. 55, no. 1, pp. 413–421, 2008.

[51] A. Nilim and L. El Ghaoui, "Robust Control of Markov Decision Processes with Uncertain Transition Matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.

[52] W. Wiesemann, D. Kuhn, and B. Rustem, "Robust Markov Decision Processes," *Mathematics of Operations Research*, vol. 38, no. 1, pp. 153–183, 2013.

[53] D. W. Park, J. Kwon, and K. M. Lee, "Robust Visual Tracking using Autoregressive Hidden Markov Model," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 1964–1971, IEEE, 2012.

[54] T. Vojir, J. Matas, and J. Noskova, "Online Adaptive Hidden Markov Model for Multi-Tracker Fusion," *Computer Vision and Image Understanding*, vol. 153, pp. 109–119, 2016.

List of Publications on the Topic of Dissertation

- Vaičiulytė J., Sakalauskas L., 2020. Recursive parameter estimation algorithm of the Dirichlet hidden Markov model. *Journal of Statistical Computation and Simulation*. Taylor & Francis Production. (**ISI Web of Science, JIF=0.767**), vol. 90:2, p. 306–323. ISSN 0094-9655. eISSN 1563-5163. https://doi.org/10.1080/00949655.2019.1679144.

- Vaičiulytė J., Sakalauskas L., 2019. Recursive estimation of multivariate hidden Markov model parameters. *Computational statistics*. Heidelberg, Springer. (**ISI Web of Science, JIF=0.680**), vol. 34, p. 1337–1353. ISSN 0943-4062. eISSN 1613-9658. https://doi.org/10.1007/s00180-019-00877-z.

- Vaičiulytė J., Sakalauskas L., 2017. Rekurentinis paslėptųjų Markovo modelių parametrų vertinimo algoritmas. *Computational Science and Techniques* (**Index Copernicus, CEEOL**). Vol 5, No 1 (2017), p. 529–540, eISSN: 2029-9966, https://doi.org/10.15181/csat.v5i1.1561.

- Vaičiulytė J., Felinskas G., 2016. Paslėptųjų Markovo modelių metodo tyrimas ir taikymas balso įrašams stenografuoti. *Jaunųjų Mokslininkų Darbai* (**Index Copernicus, CEEOL**), 1(45), 71–78. https://doi.org/10.21277/jmd.v1i45.40

ABOUT THE AUTHOR

Jūratė Vaičiulytė obtained BSc degree in 2013 in the field of Informatics, Computer Science and MCs degree in 2015 in the field of Informatics at Šiauliai University, Faculty of Mathematics and Informatics. She was a PhD student at Vilnius University Institute of Data Science and Digital Technologies from 2015 to 2019. Currently she is a working as a lector at Vilniaus Kolegija (University of Applied Sciences). Her interests include machine learning methods and tools.

Paslėptųjų Markovo modelių tyrimas ir taikymas daugiamačių sekų palaipsnei analizei

Santrauka

Tyrimų sritis ir problemos aktualumas

Svarbus įvairių taikomųjų sričių (tokių kaip kompiuterinio matymo (angl. *computer vision*) programos, šnekos atpažinimo, vaizdų analizė, Edge-AI ir kt.) bruožas yra dideli ir nuolat srautu gaunami duomenų rinkiniai. Juos galima gana efektyviai apdoroti realiu laiku taikant įvairius algoritmus. Pastaraisiais metais ypač sparčiai populiarėja ir plečiasi dirbtinio intelekto sritis, tobulinami mašininio mokymo algoritmai dėl naujų besiformuojančių skaičiavimo iššūkių. Jie susiję su duomenų interpretavimu, mokymusi ir sprendimų priėmimu realiu laiku. Tokio pobūdžio srityse, kai duomenis reikia apdoroti ir panaudoti mokymui realiu laiku, sunku pritaikyti gilaus mokymo ar tradicinio mašininio mokymo metodus, nes jiems reikia didelės statinės mokymo duomenų aibės ir pakankamų apmokymo resursų (giliam mokymui dažnai naudojami grafiniai procesoriai). Adaptyvaus (palaipsnio) mokymo ar mokymo realiu laiku (angl. *unsupervised learning*) metodai tik pradedami tyrinėti dirbtinių neuroninių tinklų metodologijoje [43, 44].

Atsirandančiose realaus laiko sistemose apdorojamus didelius duomenų srautus bandoma modeliuoti kaip stochastinius procesus. Stochastiniai procesai yra plačiai nagrinėjami inžinerijos, gamtos mokslų, socialinių mokslų, verslo ir finansų bei kitose srityse. Stochastinių procesų įvairovė yra didžiulė, apima nepriklausomus ir identiškai pasiskirsčiusius procesus, stacionarius procesus, Gauso procesus, Markovo procesus, paslėptuosius Markovo modelius (PMM) ir kt. Nepaisant plataus stochastinių procesų taikymo, vis dar išlieka daug sudėtingų uždavinių, susijusių su gebėjimu modeliuoti tikrovę. Iš tiesų, būsenų vertinimo [45–47], modelio parametrų vertinimo [1, 11, 19] ir sprendimų priėmimo [48–52] uždaviniai ir toliau nagrinėjami signalų apdorojimo, automatinio valdymo ir informacijos teorijos srityse. Įvairiuose dirbtinio intelekto modeliuose kyla panašių uždavinių, todėl išsprendus vienos srities uždavinį, galima daryti išvadas apie kitus modelius.

Pastaraisiais dešimtmečiais iškeltas modifikuotas PMM parametrų vertinimo uždavinys. Jame įvestas papildomas reikalavimas, kad stebėjimai turi būti apdorojami nuosekliai (t. y. palaipsniui), o ne saugomi kompiuterio atmintyje ir apdorojami kaip vientisas rinkinys. Ši palaipsnio PMM parametrų vertinimo formuluotė tapo didelės teorinės ir praktinės reikšmės, kadangi kai kuriose taikomosiose programose (pvz., objektų aptikimo ir stebėjimo [10, 53, 54]) skaičiavimo požiūriu neįmanoma saugoti ir apdoroti didelių stebėjimo duomenų partijų (ar rinkinių) [19, 20]. Palaipsnis PMM parametrų vertinimas taip pat svarbus programose, kuriose PMM parametrai gali kisti laike.

Baum-Welch (rinkinio) algoritmas yra skirtas ne palaipsnio PMM parametrų vertinimo uždaviniui spręsti, tačiau juo remiasi daugybė palaipsnių MVM metodų [1–5]. Kiti siūlomi PMM parametrų vertinimo metodai remiasi palaipsniais didžiausio tikėtinumo [6, 7], prognozavimo klaidų (angl. *prediction error*) metodais [7–9]. Šie palaipsniai PMM parametrų vertinimo būdai įprastai konverguoja į jų tikslo funkcijų lokalius (ne globalius) ekstremumus.

Pastaruoju metu pasiūlyti du nauji PMM būsenų perėjimo parametrų vertinimo metodai vienmačiams Gauso modeliams (su įrodytomis konvergavimo savybėmis) [10, 11], naudojant ergodines (paslėptųjų) Markovo grandinių būsenos procesų ir informacijos teorijų sąvokas. ’[10, 11] straipsnių autorių siūlomi vertinimo metodai yra neprieštaringi (angl. *consistent*) vertinant PMM būsenų perėjimo parametrus, esant PMM struktūros apribojimo sąlygoms ir turint visas žinias apie PMM stebėjimo procesą. Tačiau bendru nežinomų parametrų atveju – tiek būsenų, tiek stebėjimų procesuose – nuoseklus daugiamačių (nebūtinai Gauso) PMM parametrų vertinimas vis dar yra reikšmingas neišspręstas uždavinys. Šioje disertacijoje pristatoma nauja metodologija, kuria rinkinio algoritmams kuriami palaipsnių algoritmų atitikmenys. Nagrinėjami klasikiniai didžiausio tikėtinumo metodai yra asimptotiškai optimalūs, todėl kitokie algoritmai (pvz, dirbtinių neuroninių tinklų) iš principo negali pateikti geresnių rezultatų. Tad prasminga disertacijoje gautus rezultatus ir pasiekimus pritaikyti adaptyvių (palaipsnių) dirbtinio intelekto metodų kūrimui.

Tyrimų objektas

Disertacijos tyrimo objektas – palaipsniai algoritmai, skirti diskrečiųjų paslėptųjų Markovo modelių daugiamačiams parametrams vertinti, kai stebėjimai yra tolydieji dydžiai.

Darbo tikslas ir uždaviniai

Tikslas:

- Diskrečiųjų paslėptųjų Markovo modelių daugiamačių parametrų vertinimo palaipsnių algoritmų kūrimas, jų tyrimas ir taikymas atsitiktinių sekų analizėje, kai stebėjimai yra tolydieji dydžiai.

Uždaviniai:

- Analitiškai apžvelgti atsitiktinių sekų nuoseklioje analizėje taikomus paslėptųjų Markovo modelių parametrų vertinimo metodus.

- Sudaryti palaipsnius paslėptųjų Markovo modelių daugiamačių parametrų vertinimo algoritmus.

- Sukurtus palaipsnius paslėptųjų Markovo modelių daugiamačių parametrų vertinimo algoritmus ištirti statistinio modeliavimo būdu ir palyginti su kitais paslėptųjų Markovo modelių parametrų vertinimo algoritmais.

- Sukurtus palaipsnius paslėptųjų Markovo modelių daugiamačių parametrų vertinimo metodus pritaikyti atsitiktinių sekų nuoseklioje analizėje.

Tyrimų metodai

Šios disertacijos tyrimas pagrįstas šiais metodais:

- Darbo tikslui pasiekti ir uždaviniams spręsti analizuojami moksliniai palaipsnių PMM parametrų vertinimo algoritmų tyrimai.
- Naudojami informacijos paieškos, sisteminimo, analizės, lyginamosios analizės ir apibendrinimo metodai.
- Sukurtas palaipsnis PMM parametrų vertinimo algoritmas tiriamas Monte-Karlo metodu ir sprendžiant testinius uždavinius.
- Eksperimentinio tyrimo metodu atliekamas stebėjimų apdorojimas ir statistinė tyrimų rezultatų analizė, o gautiems rezultatams įvertinti naudojamas lyginimo ir apibendrinimo metodas.
- Darbe naudojamos algoritmų teorijos, duomenų gavybos, statistinės analizės, atpažinimo teorijos žinios.

Mokslinis darbo naujumas

Tyrimas yra moksliškai reikšmingas dėl šių priežasčių:

- Disertacijoje sukurti ir eksperimentiškai ištirti daugiamačių stebėjimų palaipsniai klasifikavimo metodai, paremti paslėptaisiais Markovo modeliais.
- Siūlomas daugiamačių Gauso paslėptųjų Markovo modelių būsenų perėjimo tikimybių palaipsnis skaičiavimo metodas, paremtas Čapmano ir Kolmogorovo lygtimi. Šiuo būsenų perėjimo tikimybių skaičiavimo metodu žinomuose palaipsniuose

algoritmuose gaunamas didesnis klasifikavimo tikslumas nei įprastine tiesioginio sklidimo (angl. *forward*) procedūra.

- Disertacijoje pateikiamas naujas palaipsnis algoritmas paslėptųjų Markovo modelių parametrams vertinti, kai stebėjimai yra pasiskirstę pagal Dirichlė skirstinį.

- Sudaryti algoritmai pritaikyti pavienių žodžių atpažinimo, užimtumo nustatymo ir pulsarų nustatymo uždaviniuose. Eksperimentų rezultatai patvirtino pasiūlytų algoritmų efektyvumą – algoritmų skaičiavimo laikas sumažėja, o klasifikavimo tikslumas nežymiai pakinta (iki 3 %).

- Palaipsnių algoritmų tyrimai gali būti taikomi kito tipo algoritmų palaipsnių atitikmenims kurti bei taikyti srautu gaunamų duomenų klasifikavimui.

Praktinė darbo reikšmė

Siūlomi palaipsniai paslėptųjų Markovo modelių daugiamačių parametrų vertinimo algoritmai gali būti naudojami įvairiose daugiamačių duomenų apdorojimo sistemose ir įrankiuose, kuriuose analizuojami duomenys yra stochastinio proceso pobūdžio, o mokymo duomenų saugojimo reikalavimai ribojami:

- palaipsnis Gauso PMM parametrų vertinimo algoritmas pritaikytas pavieniams žodžiams atpažinti, kai fiksuotas šnekos duomenų kiekis yra skirtas apmokyti, o tolimesni šnekos duomenys yra atpažįstami ir naudojami pakartotiniam parametrų vertinimui.

- palaipsnis Dirichlė PMM parametrų vertinimo algoritmas pritaikytas užimtumo nustatymo uždavinyje, kai iš sensorių gautų duomenų reikia nustatyti, ar analizuojama patalpa yra užimta, ar ne.

- palaipsnis Dirichlė PMM parametrų vertinimo algoritmas pritaikytas pulsarų kandidatų nustatymo uždavinyje.

Ginamieji teiginiai

Disertacijos ginamieji teiginiai yra šie:

- Palaipsnis PMM parametrų vertinimo algoritmas, kai išvesties tikimybinis skirstinys yra Gauso, tiesinio sudėtingumo ir klasifikavimo (stebėjimų atpažinimo) tikslumu prilygsta tradiciniam (rinkinio) Baum-Welch algoritmui.
- Taikant Čapmano ir Kolmogorovo lygtį modelio būsenų perėjimo tikimybėms skaičiuoti modelio parametrai konverguojami geriau nei su tradicine tiesioginio sklidimo (angl. *forward*) procedūra.
- Egzistuoja pakankamas duomenų rinkinys, skirtas pradinei algoritmo aproksimacijai, užtikrinantis algoritmo stabilumą ir neleidžiantis jam konverguoti į išsigimusius lokalius tikėtinumo funkcijos ekstremumus.
- Palaipsnis Gauso ir Dirichlė PMM parametrų vertinimo algoritmai gali būti taikomi kelių klasių klasifikavimo praktiniams uždaviniams, išreiškiamiems stochastiniu procesu ir modeliuojamiems paslėptaisiais Markovo modeliais, spręsti.

Darbo rezultatų aprobavimas

Pagrindiniai tyrimo rezultatai pristatyti tarptautinėse bei respublikinėse konferencijose.

Pranešimai skaityti šiose respublikinėse konferencijose:

- Konferencija „Kompiuterininkų dienų 2017“. Pranešimas „Rekurentinis paslėptųjų Markovo modelių parametro vertinimo algoritmas“. Lietuva, Kaunas, 2017 m. rugsėjo 21–22 d.

- Respublikinė mokslinė-praktinė konferencija „Informacinių technologijų iššūkiai kūrybos ekonomikoje“. Pranešimas „Paslėptųjų Markovo modelių parametrų rekurentinis vertinimas“. Lietuva, Šiauliai, 2017 m. kovo 17 d.

- Lietuvos Matematikų Draugijos 59-oji konferencija. Pranešimas „Rekurentinis paslėptųjų Markovo modelių parametrų vertinimo algoritmas ir jo taikymai“. Lietuva, Kaunas, 2018 m. birželio 18–19 d.

- Konferencija „Operacijų tyrimas ir taikymai“. Pranešimas „Paslėptųjų Markovo modelių parametrų vertinimo algoritmo tyrimas“. Lietuva, Kaunas, 2016 m. balandžio 8 d.

- Konferencija „Kompiuterininkų dienų 2015“. Pranešimas „Automatinio šnekos atpažinimo metodų tyrimas ir taikymas balso įrašams stenografuoti“. Lietuva, Panevėžys, 2015 m. rugsėjo 17–19 d.

Stendiniai pranešimai pristatyti šiose konferencijose:

- Tarptautinė 8-oji konferencija „Data Analysis Methods for Software Systems“. Stendinis pranešimas „Recurrent estimation of homogeneous Hidden Markov model parameters“. Lietuva, Druskininkai, 2016 m. gruodžio 1–3 d.

Pranešimai skaityti šiose tarptautinėse konferencijose:

- Tarptautinė konferencija EURO 2018 (29th European Conference on Operational Research). Pranešimas „Recurrent parameter estimation algorithm in hidden Markov models with application to multivariate data analysis and signal recognition“. Ispanija, Valensija, 2018 m. liepos 8–11 d.

- Tarptautinė konferencija ICIC (3rd International Conference INNOVATIONS AND CREATIVITY). Pranešimas „'Recursive Dirichlet Hidden Markov Model Parameter Estimation Algorithm“. Latvija, Liepoja, 2019 m. birželio 6–8 d.

Darbo rezultatų publikavimas

Straipsniai recenzuojamuose Lietuvos ir užsienio leidiniuose:

- Vaičiulytė J., Sakalauskas L., 2020. Recursive parameter estimation algorithm of the Dirichlet hidden Markov model. *Journal of Statistical Computation and Simulation*. Taylor & Francis Production. (**ISI Web of Science, JIF=0.767**), vol. 90:2, p. 306–323. ISSN 0094-9655. eISSN 1563-5163. https://doi.org/10.1080/00949655.2019.1679144.

- Vaičiulytė J., Sakalauskas L., 2019. Recursive estimation of multivariate hidden Markov model parameters. *Computational statistics*. Heidelberg, Springer. (**ISI Web of Science, JIF=0.680**), vol. 34, p. 1337–1353. ISSN 0943-4062. eISSN 1613-9658. https://doi.org/10.1007/s00180-019-00877-z.

- Vaičiulytė J., Sakalauskas L., 2017. Rekurentinis paslėptųjų Markovo modelių parametrų vertinimo algoritmas. *Computational Science and Techniques* (**Index Copernicus, CEEOL**). Vol 5, No 1 (2017), p. 529–540, eISSN: 2029-9966, https://doi.org/10.15181/csat.v5i1.1561.

- Vaičiulytė J., Felinskas G., 2016. Paslėptųjų Markovo modelių metodo tyrimas ir taikymas balso įrašams stenografuoti. *Jaunųjų Mokslininkų Darbai* (**Index Copernicus, CEEOL**), 1(45), 71–78. https://doi.org/10.21277/jmd.v1i45.40

Santraukos konferencijų leidiniuose:

- Vaičiulytė, Jūratė. Recurrent estimation of homogeneous Hidden Markov model parameters. *Data analysis methods for software systems : 8th international workshop on data analysis methods for software systems* [abstracts book], Druskininkai, 2016 m. gruodžio 1–3 d. Vilnius: Vilniaus universiteto leidykla, 2016. ISBN 9789986680611. p. 62.

Disertacijos struktūra

Darbą sudaro įvadas, keturi skyriai, išvados, literatūros sąrašas, autorės publikacijų disertacijos tema sąrašas.

Įvade pateikiami tyrimų sritis, objektas, darbo tikslas ir uždaviniai, tyrimų metodai, mokslinis darbo naujumas, praktinė darbo reikšmė, ginamieji teiginiai.

Pirmame skyriuje pateikiamas paslėptųjų Markovo modelių parametrų palaipsnio vertinimo metodų analitinis tyrimas, aptariamas pasirinktos temos aktualumas ir bendra problematika.

Antrame skyriuje sudaromas palaipsnis paslėptųjų Markovo modelių daugiamačių parametrų vertinimo algoritmas, paremtas didžiausio tikėtinumo metodu ir klasikiniu MVM algoritmu, ir pristatomas jo taikymas daugiamačiams stebėjimams, pasiskirsčiusiems pagal Gauso dėsnį, klasterizuoti. Aprašomi su sintetiniais duomenimis atlikti eksperimentai, norint ištirti siūlomo algoritmo savybes.

Trečiame skyriuje sudaromas palaipsnis paslėptųjų Markovo modelių daugiamačių parametrų vertinimo algoritmas, paremtas didžiausio tikėtinumo metodu, kai daugiamačiai stebėjimai yra pasiskirstę pagal Dirichlė skirstinį. Aprašomi su sintetiniais duomenimis atlikti eksperimentai, norint ištirti siūlomo algoritmo savybes.

Ketvirtame skyriuje aprašomas sudarytų Dirichlė ir Gauso paslėptųjų Markovo modelių parametrų palaipsnio vertinimo algoritmų taikymas pavienų žodžių atpažinimo, užimtumo nustatymo ir pulsarų nustatymo uždaviniuose. Aprašomas algoritmų efektyvumo tyrimas ir lyginimas su kitais egzistuojančiais algoritmais.

Disertacijos apimtis: 128 puslapiai, 29 lentelės, 23 iliustracijos. Disertacijoje remtasi 120 literatūros šaltinių.

Bendros išvados

Disertacijoje išnagrinėtas palaipsnis paslėptųjų Markovo modelių daugiamačių parametrų vertinimo uždavinys. Sudaryti algoritmai palaipsniam paslėptųjų Markovo modelių parametrų vertinimui su daugiamačiais Gauso ir ne Gauso (Dirichlė) PMM išvesties tikimybiniais skirstiniais.

Pagrindiniai rezultatai:

- Remiantis didžiausio tikėtinumo metodu ir klasikiniu MVM algoritmu, suformuoti palaipsniai PMM parametrų vertinimo algoritmai, kai daugiamačiai Gauso ir Dirichlė dėsniai yra modelio būsenų išvesties tikimybiniai skirstiniai (atitinkamai, P_GPMM_PVA ir P_DPMM_PVA).

- PMM parametrų vertinimo dalyje būsenų perėjimo tikimybes siūloma skaičiuoti remiantis Čapmano ir Kolmogorovo lygtimi. Taip pakeista tradicinė tiesioginio-atbulinio sklidimo procedūra, įprastai taikoma Baum-Welch algoritmuose, o palaipsniuose – tiesioginio sklidimo procedūra.

- Atliktas pasiūlytų palaipsnių algoritmų eksperimentinis tyrimas su sintetiniais duomenimis Monte-Karlo metodu, pavienių žodžių, užimtumo nustatymo ir pulsarų nustatymo duomenų rinkiniais.

Atlikus eksperimentinius tyrimus, suformuluotos šios išvados bei jas patvirtinantys eksperimentų rezultatai:

- Palaipsnis PMM parametrų vertinimo algoritmas, kai išvesties tikimybinis skirstinys yra Gauso, yra greitesnis ir klasifikavimo tikslumu prilygstantis tradiciniam rinkinio (Baum-Welch) algoritmui.

  – Atlikus eksperimentus gautas palaipsnio algoritmo (P_GPMM_PVA) duomenų apdorojimo greitis yra didesnis už rinkinio algoritmo. Palaipsnio algoritmo skaičiavimo laikas yra 3–9 kartus greitesnis už rinkinio algoritmo, kai apdorojami skirtingo dimensijų skaičiaus stebėjimai.
  – Rinkinio ir Palaipsnio (P_GPMM_PVA) algoritmų būsenų perėjimo tikimybių matricos, gautos PMM parametrų vertinimo metu apdorojant iki 10000 stebėjimų, nesiskiria.
  – Palaipsniu (P_DPMM_PVA) ir rinkinio algoritmais gautų PMM parametrų įverčių standartinių paklaidų santykis rodo, kad palaipsnis algoritmas, apdorodamas skirtingų dimensijų stebėjimus, duoda minimaliai nuo rinkinio algoritmo besiskiriančius parametrų įverčius. Palaipsnio ir rinkinio PMM parametrų vertinimo algoritmų standartinių paklaidų santykis svyruoja nuo 1,1 iki 1,5 karto, priklausomai nuo apdorojamų duomenų dimensijų ir sklaidos didėjimo.
  – Atliktas eksperimentinis tyrimas, kuriame lyginti palaipsnio (P_GPMM_PVA) algoritmo rezultatai su žinomu palaipsniu algoritmu. Rezultatai parodė, kad būsenų perėjimo tikimybių skaičiavimas su Čapmano ir Kolmogorovo lygtimi pagerina parametrų artėjimą prie tikrųjų modelio parametrų reikšmių, lyginant su algoritmu, kuriame būsenų perėjimo tikimybių matrica skaičiuojama su tiesioginio sklidimo procedūra. P_GPMM_PVA algoritmu apdorojus trimačius stebėjimus, standartinė PMM parametrų įverčių paklaida siekė 0,008, o žinomo algoritmo – 0,19.

- Egzistuoja pakankamas pradinis duomenų rinkinys, skirtas pradinei algoritmo aproksimacijai, kuris užtikrina algoritmo stabilumą ir neleidžia jam konverguoti į išsigimusius lokalius tikėtinumo funkcijos ekstremumus.

  - Atliktų eksperimentų, skirtų ištirti pradinės aproksimacijos dydžio įtaką atpažinimo tikslumui, gauti rezultatai parodė, kad standartinė parametrų įverčių paklaida mažėja, kai didėja bendras duomenų rinkinio dydys ir kai minimalus pradinės aproksimacijos duomenų rinkinio dydis yra pakankamas. Penkių klasterių dvimačių ir aštuonmačių stebėjimų apdorojimo atveju modelio parametrų įverčių standartinė paklaida nesiskiria, pradinio duomenų rinkinio dydis yra 300 stebėjimų ir didinamas iki 400.
  - Esant nedideliam klasterių (PMM būsenų) skaičiui, pradiniam mokymui naudojamų duomenų rinkinio dydis taip pat gali būti mažas, nepriklausomai nuo stebėjimo vektorių dimensijų skaičiaus. Tačiau didėjant klasterių skaičiui, turi didėti ir pradinės aproksimacijos duomenų rinkinys.
  - o Pritaikius palaipsnį Gauso PMM parametrų vertinimo algoritmą pavieniams žodžiams atpažinti pasiektas 97 % atpažinimo tikslumas TIDIGITS garsyno atveju, o Spoken Arabic Digits garsyno atveju atpažinimo tikslumas siekė 91 %.

- Duomenų sklaida ir dimensijų skaičius lemia palaipsnio PMM parametrų vertinimo efektyvumą – kai didėja dimensijų skaičius, didėja ir PMM parametrų įverčių standartinė paklaida – skirtumas tarp tikrųjų parametrų reikšmių ir parametrų įverčių.

  - Standartinė paklaida, kai duomenys yra dvimačiai ir sklaida – 20, yra apie 4–6 kartus didesnė už standartinę paklaidą, gautą apdorojus 16-mačius duomenis, kurių sklaida yra 50.

- Kai didėja duomenų sklaida ir dimensijos, atitinkamai didėja algoritmo ir skaičiavimo laikas.
- Kai didėja apdorojamų duomenų kiekis, modelio parametrų įverčių standartinė paklaida atitinkamai mažėja.

Atlikus eksperimentinius tyrimus su palaipsniu Dirichlė PMM parametrų vertinimo algoritmu, suformuluotos šios išvados bei jas patvirtinantys eksperimentiniai rezultatai:

- Atliktų eksperimentų, skirtų patikrinti palaipsnio Dirichlė PMM parametrų vertinimo algoritmo efektyvumą lyginant su palaipsniu Gauso PMM parametrų vertinimo algoritmu, rezultatai parodė, kad plačiai paplitę Gauso PMM negali būti taikomi uždaviniuose, kuriuose stebėjimai yra pasiskirstę pagal Dirichlė dėsnį – tokiu atveju Dirichlė PMM yra efektyvesnis ir duoda 26 % didesnį stebėjimų klasifikavimo tikslumą.

- Atlikus eksperimentus nustatyta, kad atpažinimo (klasifikavimo) tikslumas didėjant mokymo duomenų kiekiui išlieka stabilus ir nemažėja, o standartinė parametrų įverčių paklaida mažėja. Po pradinės aproksimacijos gauta PMM parametrų įverčių standartinė paklaida sumažėjo 2,5 karto pakartotinio parametrų vertinimo metu apdorojus 4000 stebėjimų.

- Pritaikius Dirichlė PMM praktiniuose uždaviniuose, gauti rezultatai parodė, kad užimtumo nustatymo duomenų rinkinio atveju pasiektas 97 % atpažinimo tikslumas, o pulsarų nustatymo duomenų rinkinio atveju atpažinimo tikslumas siekė 93 %.

- Požymių aibės modifikavimas turi įtakos atpažinimo tikslumui. Užimtumo nustatymo atveju atpažinimo tikslumas nekinta modifikavus požymių aibę – iš jos atitinkamai pašalinus „vandens garų kiekis ore“, „santykinis drėgnumas“ ir „data“ požymius. Pulsarų nustatymo atveju – modifikavus požymių aibę (iš jos pašalinus tik „DM-SNR kreivės asimetriškumo koeficientas“ požymį) nežymiai (0,01 %) pagerėja atpažinimo tikslumas.

Palaipsnių algoritmų tyrimai yra naudingi kuriant kito tipo algoritmų palaipsnius atitikmenis ir juos taikant klasifikavimui realiu laiku.

Vilniaus universiteto leidykla
Saulėtekio al. 9, LT-10222 Vilnius
El. p. info@leidykla.vu.lt,
www.leidykla.vu.lt
Tiražas 40 egz.