Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Matematyczny
specjalność: Analiza Danych

Klaudia Balcer

FlowHMM

Praca magisterska
napisana pod kierunkiem
dr. hab. Pawła Lorka

Wrocław 2023

Contents

1 Introduction 4

2 Hidden Markov Model 5

2.1 Training and Inference 6

2.1.1 Forward Algorithm 6

2.1.2 Backward Algorithm 6

2.1.3 ELBO 7

2.1.4 Forward-Backward Algorithm 7

2.1.5 Viterbi Algorithm 8

2.2 Limitations 9

3 Co-occurence based learning schema 11

3.1 Discretization 11

3.1.1 Select points at random 11

3.1.2 Quasi random 11

3.1.3 unifromly by coordinates 11

3.1.4 latin hypercude 11

4 Related work - HMM extensions 11

5 Moder approach: Flow HMM 12

5.1 Flow NN 12

5.2 Combined FlowHMM 12

5.3 Learning 12

5.3.1 Baum-Welch - adapted 12

5.3.2 Cooc-based learning - adapted 12

6 Experiments 12

6.1 Becnhmarks 12

6.2 Results 12

7 Conclusions 12

Abstract

Główny wkład własny: uogólnienie FlowHMM żeby dobrze działał z danymi wielowymiarowymi (artykuł skupia się na jednej jednowymiarowej sekwencji). Dostrojenie moedelu bedzie odbywać się w dwóch miejscach: wybór techniki dyskretyzacji oraz wybór architektury normalizing flowa odpowiednich dla wielowymiarowych danych.

1 Introduction

The history of **H**idden **M**arkov **M**odels (HMMs) dates back to 20th century[11]. However, scientist still pay interest into its further development. One of the reasons for their continuing popularity is the wide range of its application. Another advantage of this approach is its simplicity, which is becoming a significant advantage in the context of the growing importance of the explainability in the field of artificial intelligence. *TODO* ... [**?**]

In this thesis, I will present FlowHMM one of the recently proposed extension of the standard model proposed by Lorek at al.[**?**]. Extending the standard model by a flow neural network allows to model emissions with unknown distribution, which makes the model more general allows us to disregard the limiting assumption about known parametrized distributions of observed values. *TODO*...

The thesis is structured in the following manner: *TODO* ...

2 Hidden Markov Model

Standard Hidden Markov Models were first proposed by Leonard E. Baum[11] in the late 1960s and early 1970s. Shortly, the basic discrete model became extended for various continuously distributed observations and found applications in many areas such as speech recognition[13][3], biology[6][10], finance[15][16]. In this section, I will present the detailed definition of HMM for known (unspecified) distribution.

The Hidden Markov Model build up from two stochastic processes: a Markov Chain $\{X_t\}_{t\in\mathbb{N}}$ in the hidden layer, and an observable process $\{Y_t\}_{t\in\mathbb{N}}$. The model schema (for $t \in \mathbb{N}_{\leq 5}$) is presented on Figure 1.

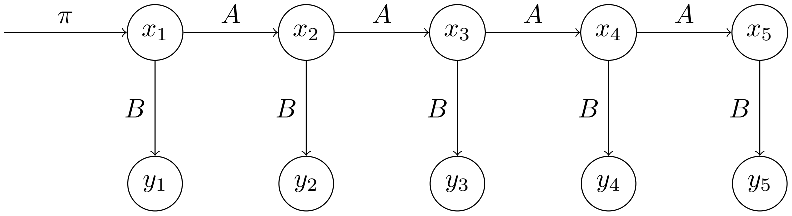

Figure 1: Schema of Hidden Markov Model.

The Markov Chain $\{X_t\}_{t\in\mathbb{N}}$ is a time-homogenuous discrete stochastic process of hidden states $\mathcal{S} = \{s_1, \ldots, s_N\}$. The process is described by a starting probability π and a transition matrix A:

$$\pi_i = \mathbb{P}(X_1 = s_i) \tag{1a}$$

$$A_{i,j} = \mathbb{P}(X_{k+1} = s_j | X_k = s_i) \tag{1b}$$

and it follows the Markov Assumption:

$$\mathbb{P}(X_{k+1} = x_{k+1} | X_{1:} = x_{1:}) = \mathbb{P}(X_{k+1} = x_{k+1} | X_k = x_k)$$

The observable process $\{Y_t\}_{t\in\mathbb{N}}$ is a sequence of random variables taking values from $\mathcal{V}$, which can be a discrete or continuous space. The observed values depend on the hidden states and follow the assumption:

$$\mathbb{P}(Y_{k+1} = y_{k+1} | Y_{1:} = y_{1:}, X_{1:+1} = x_{1:+1}) = \mathbb{P}(Y_{k+1} = y_{k+1} | X_{k+1} = x_{k+1})$$

Let us denote this probability as:

$$p(y_{k+1} | x_{k+1}) := \mathbb{P}(Y_{k+1} = y_{k+1} | X_{k+1} = x_{k+1}) \tag{2}$$

The distribution of $Y_t | X_t$ must come from a known and parametrized (for example, discrete, Gaussian, Mixture of Gaussians, Exponential, ...). We will consider a family of distributions $B = \{B_i\}_{i=1}^{N}$, where the distribution B_i denotes the observed random variable when the state is s_i:

$$Y_t|X_t = s_i \quad \sim \quad B_i \tag{3}$$

2.1 Training and Inference

The estimation of models parameters and inference are based on likelihood:

$$\mathcal{L}(Y_{1:T}) = \sum_{i=1}^{N} \pi_i p(y_1|s_i) \cdot \sum_{x_{1:T} \in \mathcal{S}^N} \prod_{t=2}^{T} p(y_t|x_i)\mathbb{P}(X_t = x_i) \tag{4}$$

The probability of observing a specific sequence is the sum of probabilities of obtaining a sequence given any possible sequence of hidden states. There are N^T possible sequences of hidden states. Before we state the training algorithm, let us specify two efficient ways of calculating the likelihood of the sequence: the forward and the backward algorithms. Both of those are based on recurrent formulas.

2.1.1 Forward Algorithm

The forward probability is specified as follows:

$$\alpha_k(i) = \mathbb{P}(Y_{1:k} = y_{1:k}, X_k = s_i)$$

and can be calculated using the recursive formula:

$$\alpha_1(i) = \pi_i p(y_1|s_i)$$

$$\alpha_k(i) = \sum_{j=1}^{n} \alpha_{k-1}(j) A(j,i) p(y_k|s_i)$$

$$\mathbb{P}(Y_{1:T} = y_{1:T}) = \sum_{i=1}^{n} \alpha_T(i)$$

2.1.2 Backward Algorithm

The backward probability is specified as follows:

$$\beta_k(i) = \mathbb{P}(Y_{k+1:T} = y_{k+1:T}|X_k = s_i)$$

and can be calculated using the recursive formula:

$$\beta_T(i) = 1$$

$$\beta_k(i) = \sum_{j=1}^{n} A_{i,j} p(y_k|s_j)\beta_{k+1}(j)$$

$$\mathbb{P}(Y_{1:T} = y_{1:T}) = \sum_{j=1}^{N} \pi_j p(y_1|s_j)\beta_1(j)$$

2.1.3 ELBO

Evidence **L**ower **Bo**und (ELBO) is a lower bound on the logarithm of the likelihood function. It uses Jensen's inequality for providing a simpler function to optimize. As the likelihood in HMM is quite complicated, in the training, we will use ELBO instead of plain likelihood.

Let's recall the formula for the likelihood of the sequence:

$$\mathcal{L}(Y_{1:T}) = \sum_{x_{1:T} \in \mathcal{S}^T} \mathbb{P}(X_{1:T} = x_{1:T}) \prod_{t=1}^{T} p(y_t | x_{1:T})$$

We can rewrite it as: *TODO: does it ne to be elaborated?*

$$\mathcal{L}(Y_{1:T}) = \sum_{i=1}^{N} \pi_i p(y_1|s_i) \cdot \prod_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} p(y_t|x_i) \mathbb{P}(X_t = x_i | X_{t-1} = x_j) \\ \mathbb{P}(X_t = x_i, X_{t-1} = x_j)$$

The logarithm of this function is called the log-likelihood:

$$\ell(Y_{1:T}) = log\left(\sum_{i=1}^{N} \pi_i p(y_1|s_i) \right) + \sum_{t=2}^{T} log\left(\sum_{i=1}^{N} \sum_{j=1}^{N} p(y_t|x_i) \mathbb{P}(X_t = x_i | X_{t-1} = x_j) \right. \\ \left. \mathbb{P}(X_t = x_i, X_{t-1} = x_j) \right)$$

Using Yensen's inequality allows us to take the sum before the logarithm:

$$\ell(Y_{1:T}) \geq \sum_{i=1}^{N} log\left(\pi_i p(y_1|s_i) \right) + \sum_{t=2}^{T} \sum_{i=1}^{N} \sum_{j=1}^{N} log\left(p(y_t|x_i) \mathbb{P}(X_t = x_i | X_{t-1} = x_j) \right. \\ \left. \mathbb{P}(X_t = x_i, X_{t-1} = x_j) \right)$$

In parameter estimation, we try to maximize ELBO (above lower bound on log-likelihood), which results in maximizing the likelihood itself.

2.1.4 Forward-Backward Algorithm

The learning algorithm for HMM (called the Forward-Backward algorithm) uses both formulas stated above. It is also called the Baum-Welch algorithm after the authors who first proposed it. This algorithm is a special case of **E**xpectation-**M**aximization (EM) algorithm[1][2]. The missing values in the case of an HMM are the hidden states. The idea of the algorithm is to iteratively:

- calculate the most probable estimators of the missing values using current parameter estimation (step E),
- update the model's parameters using the estimates of missing values (step M).

For simplicity, we will omit the index of the iteration (which indicates the current parameter estimation) in formulas in this section.

Step E

Let's denote terms in ELBO that depend on missing information:

$$\xi_k(i,j) \coloneqq \mathbb{P}(X_{k+1} = s_j, X_k = s_i) \tag{7}$$

$$\gamma_k(i) \coloneqq \mathbb{P}(X_k = s_i) \tag{8}$$

We will replace them with their estimations calculated using current parameter estimations:

$$\gamma_k(i) = \mathbb{P}(X_k = s_i) = \frac{\mathbb{P}(X_k = s_i, Y_{0:T} = y_{0:T})}{\mathbb{P}(Y_{0:T} = y_{0:T})} = \frac{\alpha_k(i)\beta_k(i)}{\sum_{j=1}^{n} \alpha_k(j)\beta_k(j)}$$

$$\xi_k(i,j) = \mathbb{P}(X_{k+1} = s_j, X_k = s_i) = \frac{\alpha_k(i)A_{i,j}p(y_{k+1}|s_j)\beta_{k+1}(j)}{\sum_{j=1}^{n} \alpha_k(j)\beta_k(j)}$$

Step M

In this step, we are looking for new parameter estimators, which maximize the ELBO formula:

$$\sum_{i=1}^{N} log\Bigg(\gamma_1(i)p(y_1|s_i)\Bigg) + \sum_{t=2}^{T}\sum_{i=1}^{N}\sum_{j=1}^{N} log\Bigg(p(y_t|x_i)\gamma_k(i)\xi_k(i,j)\Bigg)$$

2.1.5 Viterbi Algorithm

This algorithm was to inference about the hidden states. It allows us to establish which sequence of hidden states is the most probable to emit the observed sequence. Like the forward algorithm, the Viterbi algorithm is an example of dynamic programming.

First, let's denote the likelihood of the most probable sequence prefix as:

$$\nu_t(j) = \max_{x_{1:t-1}} \mathbb{P}(x_{1:t-1}, y_{1:t}, x_t = j) \tag{9}$$

We can rewrite the above recursively:

$$\nu_1(j) = \pi_j p(y_1|x_1 = j) \tag{10a}$$

$$\nu_t(j) = \max_{s \in \mathcal{S}} \nu_{t-1}(s) A_{s,j} p(y_t | x_t = j) \tag{10b}$$

$$\nu = \max_{s \in \mathcal{S}} \nu_T(s) \tag{10c}$$

We will also consider a matrix of back pointers - the most probable previous states before X_t:

$$\mathcal{X}_{1,j} = \phi \tag{11a}$$

$$\mathcal{X}_{t,j} = \arg\max_{s \in \mathcal{S}} \nu_{t-1}(s) A_{s,j} p(y_t | x_t = j) \tag{11b}$$

$$\mathcal{X} = \arg\max_{s \in \mathcal{S}} \nu_T(s) \tag{11c}$$

The best path ends in $\mathcal{X}$ and back in time follows the $\mathcal{X}_{t,j}$ back pointer terms.

2.2 Limitations

HMMs main advantage, its simplicity, is its biggest downside at the same time as it results in strict limitations. Researchers continue to develop extensions of the model to benefit from the clear structure while addressing real-world problems.

The Markov assumption for hidden states has been broken by Hidden Semi-Markov Models[12]. HSMM allows us to model also the state duration. When we go to another state, we also pick its duration, which breaks the Markov assumption (we apply it only at the moment of transiting from one state to a new one). Another way of manipulating the process of state transiting is to use a heterogeneous Markov Chain for the hidden process (the transition probabilities depend on time)[4].

Others proposed more complicated dependencies between the states in Hierarchical Hidden Markov Models[5]. Also, the output independence assumption was broken by proposing Higher-order Markov Models [16] and Recurrent Hidden Semi-Markov Models[8].

Some of the research extend also the model without modifying its core structure, for example, Sicking et al. proposed DenseHMM[14], which extends the standard model with word2vec-inspired continuous representations of each discrete value (embedding).

Also, new learning algorithms have been explored. Even if the Baum-Welch algorithm is still commonly used, some applications required improved learning algorithms. Already in the 1990s, there was a new learning schema proposed for Factorial Hidden Markov Models. Recently, an intensively explored approach to parameter estimation is co-occurrence-based learning schemes[7].

In this thesis, we will work with two limitations of the traditional model: the learning schema in which time complexity grows like the square of the sequence length and the assumption of known distribution. They were addressed by Lorek et al. with FlowHMM[9]. However, the model has been tuned to and tested on low-dimensional data. We will focus on selecting proper discretization methods and emission model architecture for a high-dimensional setup.

3 Co-occurence based learning schema

The complexity of the learning algorithm presented in the previous section depends squarely on the length of the sequence. Thus, it becomes infeasible for large data sets. Recently *TODO* [**?**] came up with an idea of using a co-occurrence matrix for the learning, which allows us to look once at the data, summarise it into the matrix and train the model with a gradient based procedure.

The co-occurrence matrix $\mathcal{Q}$ stores the probabilities of observing two values in consequtively. However, the observed values needs to be discrete (in an continuous case we have infinitly many values which makes the co-occurrence matrix infinetly large and equal zero everywhere). Thus, before presenting the details of calculating the co-occurrence matrix, we will start with discretization procedures.

3.1 Discretization

Uwaga: zależy nam na dyskretyzacjach, które nie wymagają przejrzenia całych danych, bo chcemy być od tego możliwie niezależni.

3.1.1 Select points at random

3.1.2 Quasi random

3.1.3 unifromly by coordinates

3.1.4 latin hypercude

4 Related work - HMM extensions

- Różne rozszerzenia HMMów

5 Moder approach: Flow HMM

5.1 Flow NN

5.2 Combined FlowHMM

5.3 Learning

5.3.1 Baum-Welch - adapted

5.3.2 Cooc-based learning - adapted

6 Experiments

6.1 Becnhmarks

6.2 Results

7 Conclusions

References

[1] Leonard E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, 1967.

[2] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970. Publisher: Institute of Mathematical Statistics.

[3] Akshay Madhav Deshmukh. Comparison of Hidden Markov Model and Recurrent Neural Network in Automatic Speech Recognition. *European Journal of Engineering and Technology Research*, 5(8):958–965, August 2020. Number: 8.

[4] Jose G Dias, Jeroen K Vermunt, and Sofia Ramos. Heterogeneous Hidden Markov Models.

[5] Shai Fine, Yoram Singer, and Naftali Tishby. The Hierarchical Hidden Markov Model: Analysis and Applications. *Machine Learning*, 32(1):41–62, July 1998.

[6] Anders Krogh, Michael Brown, I. Saira Mian, Kimmen Sjölander, and David Haussler. Hidden Markov Models in Computational Biology: Applications to Protein Modeling. *Journal of Molecular Biology*, 235(5):1501–1531, February 1994.

[7] Balaji Lakshminarayanan and Raviv Raich. Non-negative matrix factorization for parameter estimation in hidden Markov models. In *2010 IEEE International Workshop on Machine Learning for Signal Processing*, pages 89–94, August 2010. ISSN: 2378-928X.

[8] Hao Liu, Lirong He, Haoli Bai, Bo Dai, Kun Bai, and Zenglin Xu. Structured Inference for Recurrent Hidden Semi-markov Model. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 2447–2453, Stockholm, Sweden, July 2018. International Joint Conferences on Artificial Intelligence Organization.

[9] Paweł Lorek and Rafał Nowak. FlowHMM: Flow-based continuous hidden Markov models.

[10] Mohammadreza Momenzadeh, Mohammadreza Sehhati, and Hossein Rabbani. Using hidden Markov model to predict recurrence of breast cancer based on sequential patterns in gene expression profiles. *Journal of Biomedical Informatics*, 111:103570, November 2020.

[11] Bhavya Mor, Sunita Garhwal, and Ajay Kumar. A Systematic Review of Hidden Markov Models and Their Applications. *Archives of Computational Methods in Engineering*, 28(3):1429–1448, May 2021.

[12] Kevin P Murphy. Hidden semi-Markov models (HSMMs).

[13] J. Picone. Continuous speech recognition using hidden Markov models. *IEEE ASSP Magazine*, 7(3):26–41, July 1990. Conference Name: IEEE ASSP Magazine.

[14] Joachim Sicking, Maximilian Pintz, Maram Akila, and Tim Wirtz. DenseHMM: Learning Hidden Markov Models by Learning Dense Representations, December 2020. arXiv:2012.09783 [cs, stat].

[15] Tak Kuen Siu. A Hidden Markov-Modulated Jump Diffusion Model for European Option Pricing. In Rogemar S. Mamon and Robert J. Elliott, editors, *Hidden Markov Models in Finance: Further Developments and Applications, Volume II*, International Series in Operations Research & Management Science, pages 185–209. Springer US, Boston, MA, 2014.

[16] Mengqi Zhang, Xin Jiang, Zehua Fang, Yue Zeng, and Ke Xu. High-order Hidden Markov Model for trend prediction in financial time series. *Physica A: Statistical Mechanics and its Applications*, 517:1–12, March 2019.