

ANALYSIS OF 2D PROBLEM IN HMM

Janusz Bobulski

Czestochowa University of Technology, Institute of computer and Information Science

janusz.b@icis.pcz.pl

Abstract: Hidden Markov models are widely applying in data classification. They are using in many areas where 1D data are processing. In the case of 2D data, appear some problems with applying 2D HMM. This paper describe the important limitations of HMM when we have to processing two dimensional data.

Keywords: image processing, hidden Markov models HMM, pseudo 2DHMM, 2DHMM

1. Introduction

Hidden Markov models are widely applying in data classification. They are using in speech recognition, character recognition, 2-D shape classification, biological sequence analysis, financial data processing, texture analysis, face recognition, etc. This widely applying HMM is result of its effectiveness. When we work with one dimensional data, we have good tools and solution for this. But when we process two dimensional data, we should apply two dimensional HMM. There is problems, because there aren't good and efficient solution of three basic problems of 2D HMM [1, 2]:

1. Given observation $O=\{O_1, \dots, O_T\}$ and model $\lambda = (A, B, \pi)$; efficiently compute $P(O| \lambda)$:
 - Hidden states complicate the evaluation
 - Given two models λ_1 and λ_2 , this can be used to choose the better one.
2. Given observation $O=\{O_1, \dots, O_T\}$ and model $\lambda = (A, B, \pi)$ find the optimal state sequence $q = (q_1, q_2, \dots, q_T)$:
 - Optimality criterion has to be decided (e.g. maximum likelihood)
 - "Explanation" for the data.
3. Given $O=\{O_1, \dots, O_T\}$; estimate model parameters $\lambda = (A, B, \pi)$ that maximize $P(O| \lambda)$.

2. Classic HMM

HMM is used to the identification process. A HMM is a double stochastic process with underlying stochastic process that is not observable (hidden), but can be observed through another set of stochastic processes that produce a sequence of observation. Let $O=\{O_1, \dots, O_T\}$

be the sequence of observation of feature vectors, where T is the total number of feature vectors in the sequence. The statistical parameters of the model may be defined as follows [3].

- The number of states of the model, N (Fig.1)
- The transition probabilities of the underlying Markov chain, $A=\{a_{ij}\} 1 \leq i, j \leq N$ where a_{ij} is the probability of transition from state i to state j subject to the constraint $\sum_{j=1}^N a_{ij} = 1$
- The observation probabilities, $B=\{b_j(O_T)\}, 1 \leq j \leq N, 1 \leq t \leq T$ which represents the probability of the t th observation conditioned on the j th state.
- The initial probability vector, $\Pi = \{\pi_i\} 1 \leq i \leq N$.

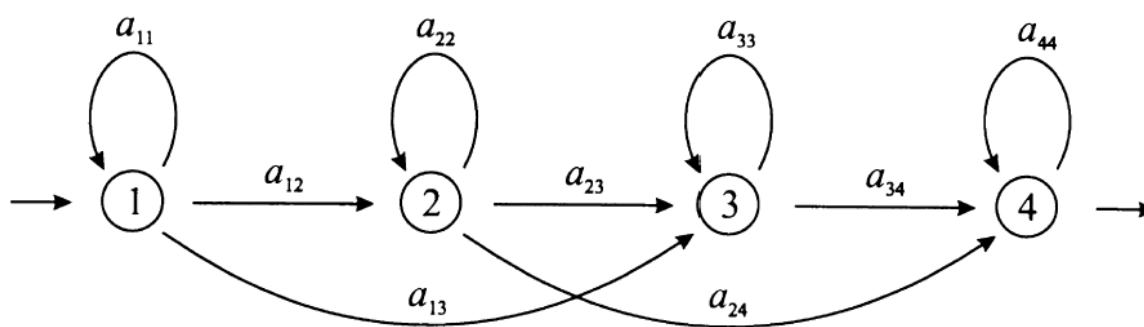


Fig.1. One-dimensional HMM [7]

Hence, the HMM requires three probability measures to be defined, A, B, π and the notation: $\lambda = (A, B, \pi)$ is often used to indicate the set of parameters of the model.

The parameters of the model are generated at random at the beginning. Then they are estimated with Baum-Welch algorithm, which is based on the Forward-Backward algorithm. Second way to estimate of parameters is Viterbi algorithm, which is very similar to Forward-Backward algorithm. The forward algorithm calculates the coefficient $\alpha_t(i)$ (probability of observing the partial sequence (o_1, \dots, o_t) such that state q_t is i). The backward algorithm calculates the coefficient $\beta_t(i)$ (probability of observing the partial sequence (o_{t+1}, \dots, o_T) such that state q_t is i). The Baum-Welch algorithm, which computes the λ , can be described as follows [1].

1. Let initial model be λ_0
2. Compute new λ based on λ_0 and observation O
3. If $\log (P(O/\lambda) - \log P(O/\lambda_0) < DELTA$ stop
4. Else set $\lambda_0 \leftarrow \lambda$ and goto step 2.

The parameters of new model λ (1), based on λ_0 and observation O , are estimated from equation of Baum-Welch algorithm (Fig.2) [1], and then are recorded to the database.

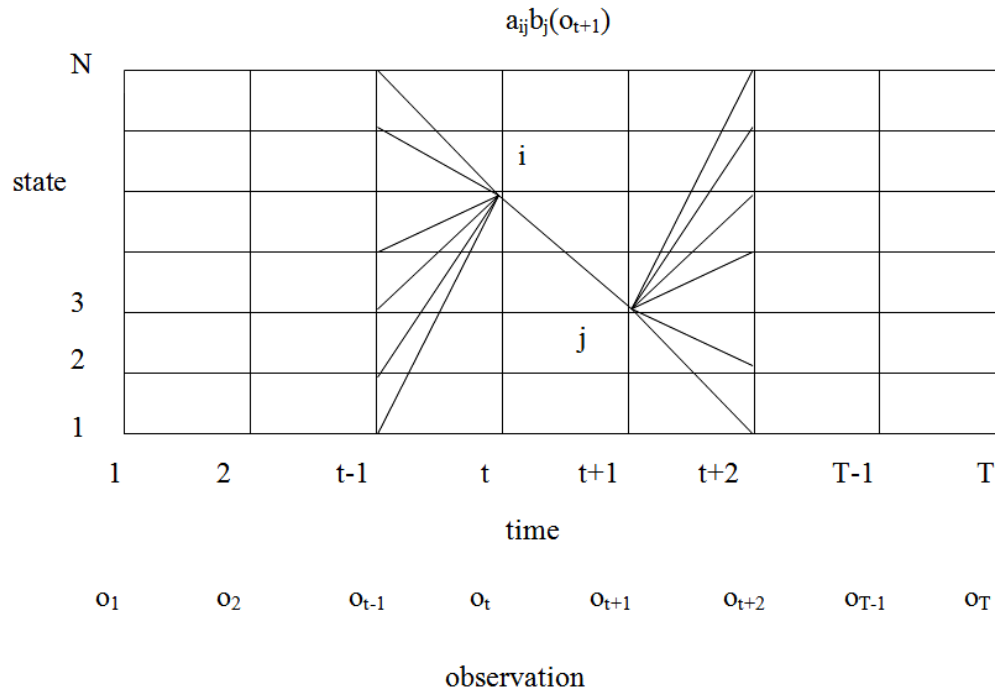


Fig.2. Baum-Welch algorithm [1].

Baum-Welch algorithm (forward-backward)

Forward probability $\alpha_j(t)$ for $2 \leq j \leq N-1$ or $1 \leq t \leq T_r$, is calculated with following recurrent formula:

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1) a_{ij} \right] b_j(o_t^r)$$

initial condition:

$$\alpha_1(1) = 1$$

$$\alpha_j(1) = a_{1j} b_j(o_1^r) \quad \text{dla } 2 \leq j \leq N-1$$

finishing condition:

$$\alpha_N(T_r) = \sum_{j=2}^{N-1} \alpha_j(T_r) a_{jN}$$

Backward probability $\beta_i(t)$ for $2 \leq i \leq N-1$ and $1 \leq t \leq T_r$, is calculated with following recurrent formula:

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(o_{t+1}^r) \beta_j(t+1)$$

initial condition: $\beta_i(T_r) = a_{iN} \quad 1 \leq i \leq N$

finishing condition: $\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(o_1) \beta_j(1).$

Problem:

For giving observation vector $O = (o_1, o_2, \dots, o_T)$ estimation model parameters $\lambda = (\pi, A, B)$ in order to take maximum $P(O | \lambda)$.

Problem solution:

- Estimate the parameters of the model $\lambda = (\pi, A, B)$ for maximum $P(O | \lambda)$,
- Define $\xi(i, j)$ as a probability of being in state i at time t and in the state j at time $t + 1$,

$$\xi(i, j) = \frac{\alpha(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} = \frac{\alpha(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}$$

Relation in algorithm:

- define $\gamma_t(i)$ as a probability being in state i at time t , given the observation sequence

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j)$$

- $\sum_{t=1}^T \gamma_t(i)$ is the expected number of times state i is visited,
- $\sum_{t=1}^{T-1} \xi_t(i, j)$ is the expected number of transition from state i to state j ,
- $\pi_i =$ expected frequency in state i at time $(t = 1) = \gamma_1(i)$,
- $a_{ij} =$ (expected number of transition from state i to state j) / (expected number of transition from state i):

$$a_{ij} = \frac{\sum \xi_t(i, j)}{\sum \gamma_t(i)}$$

- $b_j(k) =$ (expected number of times in state j and observing symbol k) / (expected number of times in state j):

$$b_j(k) = \frac{\sum_{t, o_t=k} \gamma_t(j)}{\sum_t \gamma_t(j)}$$

Viterbi algorithm [4, 5] (Fig.3)

- Define $\delta(i)$ – the highest probability path ending in state i ,

Algorithm:

- Initialisation:

$$\delta_1(i) = \pi_i b_i(o_1) \quad 1 \leq i \leq N$$

$$\psi_1 = 0$$

- recursion:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(o_t)$$

$$\psi_{t(j)} = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \quad 2 \leq t \leq T, \quad 1 \leq j \leq N$$

- termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]$$

$$q_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)]$$

- path (state sequence) backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T-1, T-2, \dots, 1$$

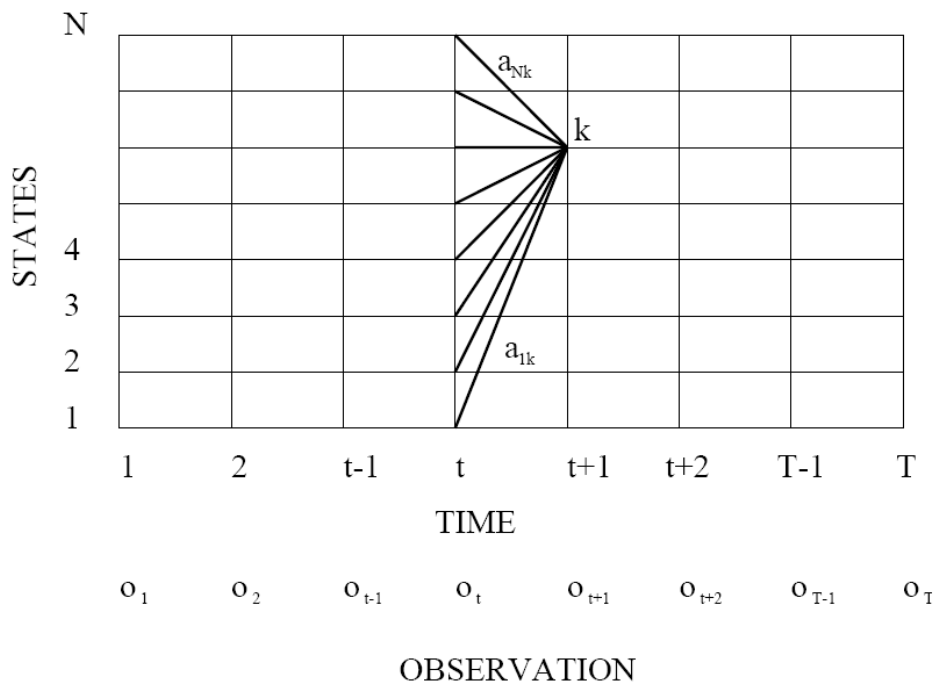


Fig.3. Viterbi algorithm [1].

The testing process consists of computing the probability of observation generating by the models saved in database and choosing this model for which the likelihood is maximum.

In the proposed method, probabilities are calculated separately for each of the three models representing parts of the face, then they are added. The face, for which the sum of probability is maximum, is chosen as the correct face. The probability of generating sequences of observations is computed from the equations (2)-(4) [1].

$$P(O | \lambda) = \sum_q P(O | q, \lambda) P(q | \lambda)$$

$$P(O | q, \lambda) = \prod_{i=1}^T P(o_i | q_i, \lambda) = b_{q_1}(o_1) b_{q_2}(o_2) \dots b_{q_T}(o_T)$$

$$P(q | \lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T}$$

3. Pseudo 2DHMM

A pseudo 2D Hidden Markov models are extension of the 1D HMM. A P2DHMM consist of a number of superstates. The topology of superstate model is a linear model, where only self transition and transition to the following superstate are possible. Inside the superstates there are linear 1D HMM. The state sequences in the rows are independent of the state sequences of neighboring rows [6, 7].

Figure 4 shows a pseudo two dimensional hidden Markov model which consist of four superstates. Each superstate contains three states one dimensional HMM.

4. 2DHMM

An extension of the HMM to work on two-dimensional data is 2D HMM (Fig.5). The principle of two dimensional hidden Markov models was described in the paper [2]. A 2D HMM can be regarded as a combination of one state matrix and one observation matrix, where transition between states take place according to a 2D Markovian probability and each observation is generated independently by the corresponding state at the same matrix position. It was noted that the complexity of estimating the parameters of a 2D HMMs or using them to perform maximum a posteriori classification is exponential in the size of data. Similar to 1D HMM, the most important thing for 2D HMMs is also to solve the three basic problems, namely, probability evolution, optimal state matrix and parameters estimation. Li Yujian in [2] proposed some analytic solution of this problems. But this solution has some disadvantages. First, the computation of parameters and probability are very complexity [8]. Second, This solution can be applying only for left-right type of HMM. And third, we can use

only small size of HMM. 2D HMM is still limited by the computational power of the machine.

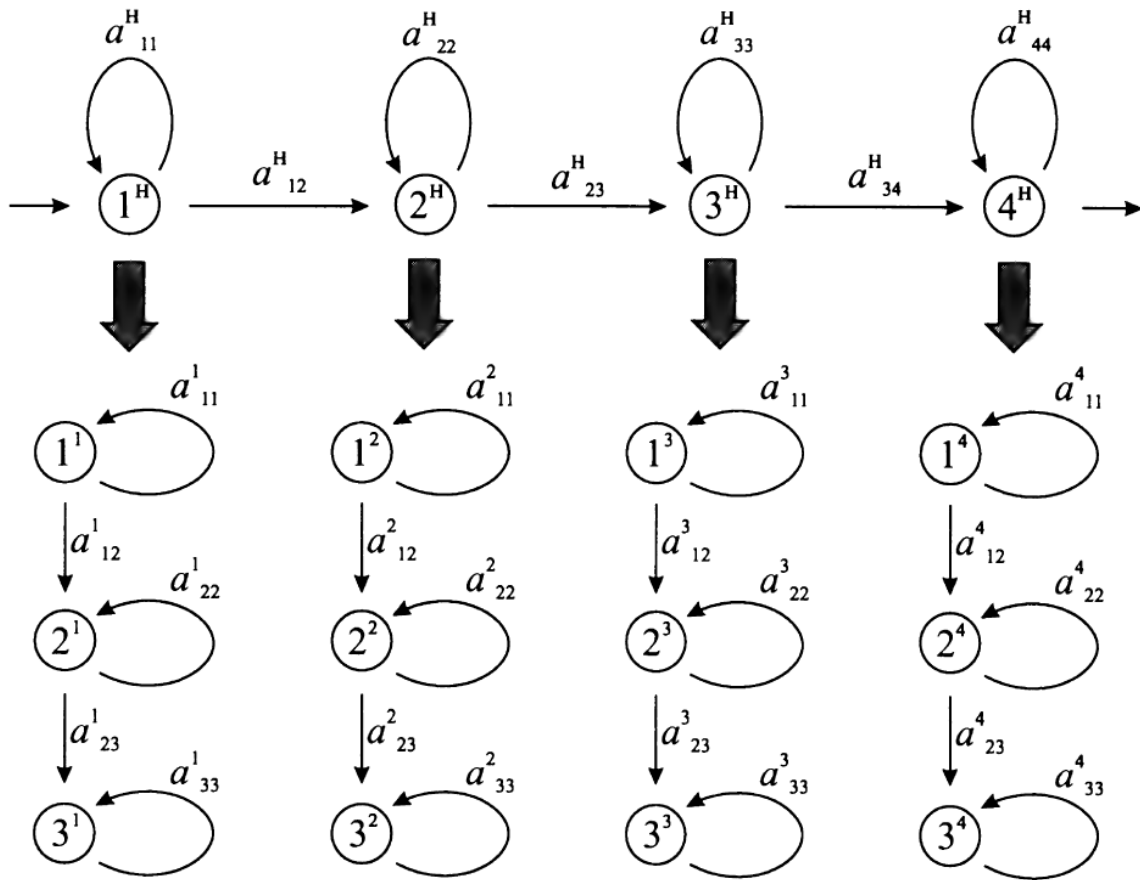


Fig.4. Pseudo 2D-HMM [6].

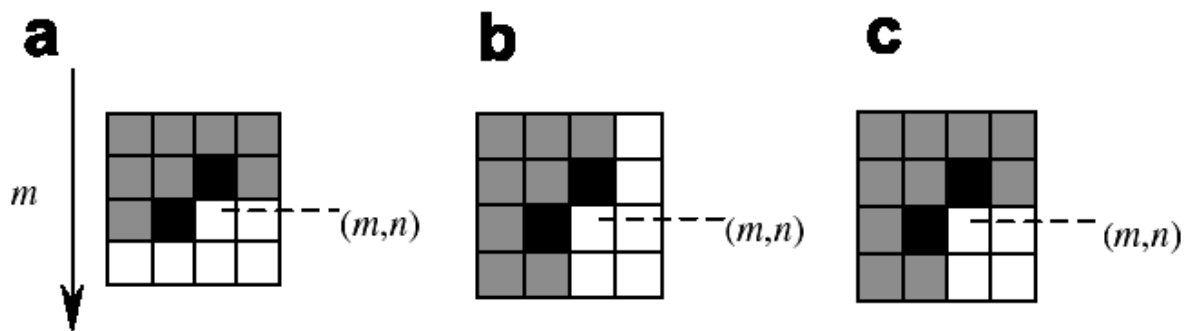


Fig.5. 2D Markovian transitions among states [2].

Conclusion

We can applying three approach to the 2D data analysis:

- reduce dimensionality data to 1D vector and use 1D HMM,
- divide data to segments and use pseudo 2D HMM
- use complexity analytic calculation in 2D HMM.

Presented solution 2D HMM is assumption of real full 2D HMM. Therefore, is needed future work on two dimensional hidden Markov models. Future solution have to resolve the three basic problems of HMM for ergodic and larger set of states and data.

References

- [1] Kanungo T., 'Hidden Markov Model Tutorial'. www.cfar.umd.edu/~kanungo, 1999
- [2] Yujian Li, An analytic solution for estimating two-dimensional hidden Markov models, *Applied Mathematics and Computation* 185(2007), pp.810-822
- [3] Rabiner L. R., A tutorial on hidden Markov models and selected application in speech recognition?. *Proc. IEEE* 77, 1989, pp. 257-285.
- [4] Hu J., Brown M.K., Turin W.: HMM Based On-Line Handwriting Recognition, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 18 No.10 (1996) 1039-1045
- [5] Forney G.D.: The Viterbi Algorithm, *Proc. IEEE*, Vol. 61 No. 3 (1973) 268-278
- [6] Eickeler Stefan, Müller Stefan, Rigoll Gerhard, High Performance Face Recognition Using Pseudo 2-D Hidden Markov Models, *European Control Conference*, <http://citeseer.ist.psu.edu>, 1999
- [7] Eickeler Stefan, Müller Stefan, Rigoll Gerhard, Recognition of JPEG compressed faceimages based on statistical methods, *Image and Vision Computing* 18(2000), pp. 279-289
- [8] Li j., Najmi A., Gray R.M., Image classification by a two dimensional Hidden Markov model, *IEEE Trans. Signal Process.* 48(2000), pp. 517-533