THREE TECHNIQUES FOR STATE ORDER ESTIMATION OF HIDDEN MARKOV MODELS

Predrag Pucar and Mille Millnert

Department of Electrical Engineering
Linköping University
S-581 83 Linköping
Sweden
Email:predrag@isy.liu.se

ABSTRACT

In this contribution three examples of techniques that can be used for state order estimation of hidden Markov models are given. The methods are also exemplified using real laser range data, and the computational burden of the three methods is discussed. Two techniques, Maximum Description Length and Maximum *a Posteriori* Estimate, are shown to be very similar under certain circumstances. The third technique, Predictive Least Squares, is novel in this context.

1 INTRODUCTION

A phenomenon that often occurs in the segmentation process is the spurious jumping in the state estimate of a hidden Markov model (HMM) when more states than needed are used. The reason for that is that the algorithms use all available degrees of freedom, *i.e.*, the algorithms actually segment the signal/image into M segments if the signal model's underlying Markov chain has M states. There is obviously a need for estimation of the number of states before applying the segmentation routine.

Example 1.1 *Assume that a white noise sequence, depicted in Fig. 1, is given. The natural choice for the number of states to model the white noise sequence is 1, since there are no jumps in the signal. If we, however, choose a two-state Markov chain and apply the Baum-Welch algorithm to segment the signal into two segments the result is the one found in Fig. 1.*

The paper is organized as follows. First a problem formulation and a motivation for looking into this kind of issues, is given. Then the three algortihms are presented, and finally a example including data from a laser range radar system, is presented.

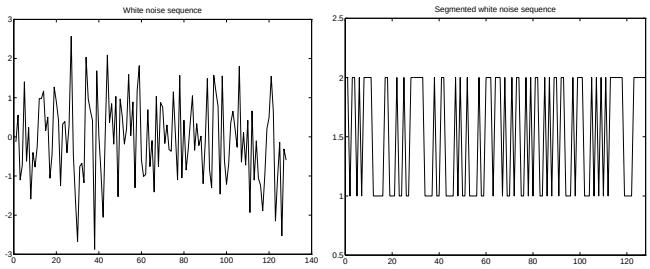

Figure 1: Left: White noise sequence with variance 1. Right: Resulting segmentation of the white noise signal using two states.

2 PROBLEM FORMULATION

We will first introduce the concept of *hidden Markov models* (HMM).

Definition 1 *An HMM is a doubly stochastic process with one underlying process that is not observable, but can only be observed through another set of output processes that produce the observed data. The underlying hidden process is a Markov chain.*

The HMMs are being used extensively in a variety of areas. The standard issue is how to estimate the parameters in the model producing the output and how to estimate the unobserved Markov chain sequence.

There is a vast literature on the above mentioned topic, see for example [1, 5]. An often circumvented problem is how to decide on how many states to use in the assumed hidden Markov chain. In practice, when one is confronted with , e.g., a segmentation problem, that kind of information is seldom known. However, it is, crucial for the result of the applied algorithm.

3 THREE ALGORITHMS

In this section the three proposed algorithms are presented. The complete derivation of the expressions will be left out in this paper. For a complete version of the paper see [2]. The hidden Markov sequence will be denoted by $z_{t_1}^{t_2}$, meaning the sequence from time instant t_1 to t_2. The subscript is suppressed when $t_1 = 1$, and the superscript is suppressed when $t_2 = t_1$. The observed process is denoted by $y_{t_1}^{t_2}$.

3.1 Minimum Description Length

Assume a sequence y^N is given and we know it has been generated by a finite state source, but we do not know the number of states M°. In the sequel M° will denote the "true" value of the model state order, M is the auxiliary variable denoting the model state order which is tested by the algorithm and $\hat{M}$ the estimate of the model state order. Usually a criterion is calculated for different values of M and then an $\hat{M}$ is chosen as an estimate. The desired result is, of course, that $\hat{M} = M^\circ$.

Assume that for every M we have a code Φ_M. A code can be described as a mapping from the source symbols to a series of bits. The mapping takes into account the distribution of the symbols. All the information theoretic techniques boil down to finding an appropriate code Φ_M for coding the sequence y^N, calculating the code length for different codes and then picking the M for which the code Φ_M gives the shortest code length when coding y^N. We have chosen the minimum description length (MDL) [3] as coding principle. Shortly, the MDL principle can be summarized as choosing the model that minimizes the number of bits it takes to describe the given sequence. Note that not only the data are encoded, using the model, but also the model itself, *i.e.*, the real-valued parameters in the model. How does this apply in the HMM case? The overall number of bits will be the sum of the number of bits for describing the data and the model. If the number of parameters in the chosen model is denoted by d and M is the number of states the following expression is obtained

$$V = \log_2\left(\frac{1}{N}\sum_{i=1}^{N} e_i^2\right) + (d + M(M-1) + 1)\frac{\log_2 N}{N}.$$

The expression above has to be calculated for different M, and the state order estimate is the M which gives minimal V.

3.2 Predictive Least Squares

The predictive least squares (PLS) idea originates from [4]. We start with a basic regression problem. Assume two sets of observations y^N and $x^N(i)$, where $i = 1, \ldots, M$, are given. The usual procedure when applying least squares is to introduce a model class and to pick a predictor for y_t. The predictor is denoted by $\hat{y}_t(\theta; y^{t-1})$. The ideal predictor should then minimize

$$E_\theta(y_t - \hat{y}_t(\theta; y^{t-1}))^2. \tag{1}$$

If the expectation in (1) is replaced with a sample mean the following estimate is obtained

$$\hat{\theta} = \arg\min_\theta \frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t(\theta))^2. \tag{2}$$

Note that the estimate of θ is based on the *whole* data set. The PLS approach is to change the predictor to $\hat{y}_t(\theta_{t-1}; y^{t-1})$, *i.e.*, at every time instant the parameter vector θ minimizing the criterion (2) is calculated using past data only. The parameter vector estimate will vary in time, since the number of data on which it is based, grows. If then all the prediction errors are accumulated we the following criterion is obtained

$$V_{PLS}(M) = \sum_{t=1}^{N}(y_t - \hat{y}_t(\theta_{t-1}; y^{t-1}))^2,$$

where M is the number of regressors x included.

In the HMM case we first have to calculate the one step predictor and then go through the PLS procedure for the HMM case. We also point out difficulties in proving consistency, although in simulations the method shows good results. The procedure is to use the EM algorithm, see for example [1], to estimate the state sequence and the probabilities $\alpha_i(t) = P(z_t|Y^t)$, where

z_t is the state of the Markov chain at time instant t, and Y^t is the data sequence up to and including time instant t. The prediction $\hat{y}_{t+1} = E\{y_{t+1}|y^t\}$ and can be calculated as follows

$$\begin{aligned} P(y_{t+1}|y^t) &= \sum_i P(y_{t+1}|y^t, z_t = i)P(z_t = i|y^t) \\ &= \sum_j \sum_i P(y_{t+1}|z_{t+1} = j, y^t, z_t = i) \cdot \\ &\quad P(z_{t+1} = j|z_t = i, y^t)P(z_t = i|y^t) \\ &= \sum_j \sum_i P(y_{t+1}|z_{t+1} = j, y^t)q_{ij}\alpha_i(t), \end{aligned}$$

where q_{ij} is the transition probabilities for the hidden Markov chain. Taking expectation of the variable $\{y_{t+1}|y^t\}$ results in

$$\hat{y}_{t+1} = \sum_j \sum_i E\{y_{t+1}|z_{t+1} = j, y^t\}q_{ij}\alpha_i(t), \qquad (3)$$

where the expectation usually is straightforward to calculate.

Since we do not know anything about the behavior of the PLS-criterion as a function of M we have to adopt an *ad hoc* rule when actually searching for the minimum of the criterion. The procedure of calculating the PLS-criterion for different model state orders M is rather computationally costly. For every M a new EM algorithm has to be run.

The procedure when using the EM algorithm and PLS is the following:

1. Decide which model state orders that are to be tested.
2. Decide what search strategy to use when testing different number of states.
3. Run the EM algorithms in accordance to the decided strategy testing the different state orders.
4. Sum the "honest" prediction errors.
5. Chose the state order that gives the lowest accumulated cost.

In step two, with the word "strategy" we mean the order in which the different EM algorithms for different model state orders should be tested.

In step four and five, at time instant t the EM algorithms are run on the data up to t and y_{t+1} is predicted according to (3). The squared errors $\varepsilon^2_{t+1}(M) = (y_{t+1} - \hat{y}_{t+1}(M))^2$ are summed up and finally when the row is completely processed we choose the number of states to equal the number of states of the model which minimized the PLS criterion.

How to choose the number of states to test is an intricate question. In our simulations we have chosen an *ad hoc* solution, we simply start from one state, and then increase the number of states by one until the PLS criterion stops to decrease and starts to increase. The usual behavior of the PLS criterion for different M is a rapid drop when we increase M and then when M passes the right number of states, *i.e.*, $M > M^\circ$, the PLS criterion starts to increase slowly. As the estimate we simply choose the value of M if the PLS criterion starts to increase for $M + 1$. The drawback of this procedure is that some a priori knowledge about the number of states is needed to avoid numerous testings. In our application we know that usually the number of states are one or two. It is very unlikely that we will need more than four states. This knowledge, of course, influences our testing strategy (start with one state and then increase the number by one). General advice is difficult to give.

Example 3.1 *In this example the PLS-method for model state order estimation is applied to a synthetic signal. We first generate a sequence of states from a three-state Markov chain. Noise is then added according to the following relation*

$$y_t = z_t + 0.1e_t,$$

where e_t is zero mean and Gaussian white noise with variance 1.

If then the previously described PLS procedure is applied in state order to estimate the number of states of the Markov chain we obtain the following accumulated error shown in Fig. 2.

The behavior of the PLS-criterion in Example 3.1 is typical. The quick drop when increasing the model state order towards the true one. After the true model state order is passed the trend is not so obvious. Depending on the realization and if short data sets are used the model state order can be overestimated.

Consistency of the Estimate One important question regarding the estimate is, of course, the convergence of the estimate when the number of data tends

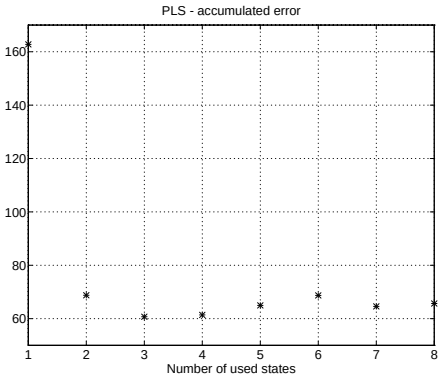

Figure 2: Resulting accumulated error obtained when using PLS and an increasing number of states of the hidden Markov chain. The minimum is obtained for three states which is in accordance with the problem statement.

to infinity. This question proves to be very difficult to answer and we have not arrived at a satisfactory treatment of that matter.

3.3 Maximum a Posteriori Estimate

The last approach is based on using a bank of Kalman filters when estimating the model parameters of the model behind the observed data, and the state sequence. The Kalman filters also give the distribution of the estimates, so for example the distribution of the data assuming no underlying Markov chain is given by the following expression

$$P(y^N) = (2\pi)^{N/2} \left(\prod_{i=1}^{N} \det S_t \right)^{-1/2} e^{-\frac{1}{2}V_N},$$

where S_t is given by the following equations

$$\begin{aligned} S_t &= \varphi^T P_{t-1} \varphi_t + \Lambda_t \\ P_t &= P_{t-1} - P_{t-1}\varphi_t S_t^{-1} \varphi_t^T P_{t-1}, \end{aligned}$$

V_N is the normalized sum of prediction errors and φ is a known vector.

When a Markov chain with M states is introduced, and after some calculations, the following expression for the likelihood of the data is obtained

$$-\frac{2}{N}\log P(y^N|z^N, M) \approx \sum_{i=T_1}^{T_M} \frac{V_{N(i)}}{N} + \sum_{i=T_1}^{T_M} \frac{d(i)\log N(i)}{N}. \quad (4)$$

In the expression above $d(i)$ denotes the number of parameters of the output process model corresponding to different Markov chain states, T_i denotes the set of time instants where the Markov chain is in state i and $N(i)$ denotes the number of elements in T_i. The result is striking in its similarity with Rissanen's MDL criterion. If we have prior knowledge of the transition matrix Q, or maybe have it as a design parameter, we can calculate the *a posteriori* probability for the states in a straightforward way.

4 EXAMPLE

In this section the MDL approach is tested using an image obtained by a laser range radar. The pixel values are the distance to the terrain measured by a laser. The objective with the segmentation algorithm, in this case the EM algorithm, is to find objects in the image that differ from the background, in other words a first step towards object recognition. The test image that is used here shows a shield in the middle of the image, and in the upper right corner there are some bushes. The way to interpret the segmented image is to look at connected areas with the same segment label, and then do further investigation by taking the estimated parameters of the observed model, variance of the residuals, *etc*, into account. The problem we are stressing here is that usually the user has to pick the number of hidden states of the Markov chain for each row (or fix one for all rows) since the image is segmented row by row. Here we used the above proposed MDL loss function. Similar results are obtained by using the MAP loss function. The estimation routine, however, is different in that case. In Fig. 3 the original laser image and the resulting segmentation is shown. Note that in the area "in front" of the shield only one hidden state is used, and that way spurious jumping as in Fig. 1 is avoided.

5 CONCLUSIONS

Three different algorithms for state order estimation of hidden Markov models are compared. The performance and computational complexity of each algorithm is investigated. In the paper it is shown under what circumstances MDL and the MAP estimate coincide.

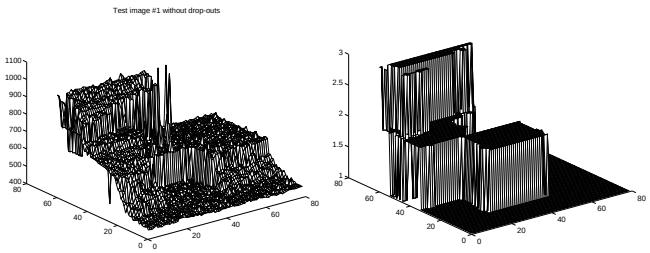

Figure 3: Left: The raw data obtained from the laser system. The z-axis is the distance to the terrain. Right: Resulting segmentation of the laser range radar image using EM and the MDL strategy.

References

[1] B.H. Juang and L.R. Rabiner. "Mixture Autoregressive Hidden Markov Models for Speech Signals". *IEEE Trans. on ASSP*, 33(6):1404–1413, December 1985.

[2] P. Pucar. Segmentation of laser range radar images using hidden markov field models. Linköping studies in science and technology. thesis no.403, liutek-lic-1993:45, isbn 91-7871-184-3, Department of Electrical Engineering, Linköping University, Sweden, 1993.

[3] J. Rissanen. "Modeling by Shortest Data Description". *Automatica*, 14:465–471, 1978.

[4] J. Rissanen. "A Predictive Least-Squares Principle". *IMA Journal of Math. Control & Information*, 3:211–222, 1986.

[5] R.G. Whiting. *Quality Monitoring in Manufacturing Systems: A Partially Observed Markov Chain Approach*. PhD thesis, University of Toronto, Canada, 1985.