Chapter 5 - Forecasting, decoding and state prediction

02433 - Hidden Markov Models

Martin Wæver Pedersen, Henrik Madsen

Course week 5

MWP, compiled June 7, 2011

LUNDS UNIVERSITET
Lunds Tekniska Högskola

Interreg IVA
ÖRESUND – KATTEGAT – SKAGERRAK

DTU

Chapter 5 - Outline

The HMMs considered in chapter 5 are not necessarily stationary, $\boldsymbol{\delta}$ is therefore the initial distribution.

Main results:

Conditional distributions:	$\Pr(X_t = x \vert \mathbf{X}^{(-t)} = \mathbf{x}^{(-t)}) = \sum_i w_i(t) p_i(x)$.
Forecast distributions:	$\Pr(X_{T+h} = x \vert \mathbf{X}^{(T)} = \mathbf{x}^{(T)}) = \sum_i \xi_i(h) p_i(x)$.
Local decoding:	$\Pr(C_t = i \vert \mathbf{X}^{(T)} = \mathbf{x}^{(T)}) = \alpha_t(i)\beta_t(i)/L_T$.
Global decoding:	$\Pr(\mathbf{C}^{(T)} = \mathbf{c}^{(T)} \vert \mathbf{X}^{(T)} = \mathbf{x}^{(T)})$ maximized over $\mathbf{c}^{(T)}$.
State prediction:	$\Pr(C_{T+h} = i \vert \mathbf{X}^{(T)} = \mathbf{x}^{(T)}) = \boldsymbol{\alpha}_T \boldsymbol{\Gamma}^h(,i)/L_T$.

Conditional distribution

Define $\mathbf{X}^{(-t)} \equiv (X_1, \ldots, X_{t-1}, X_{t+1}, \ldots, X_T)$.

Consider then the conditional distribution of an observation given the remaining observations:

$$\Pr(X_t = x | \mathbf{X}^{(-t)} = \mathbf{x}^{(-t)}) = \frac{\Pr(\mathbf{X}^{(T)} = \mathbf{x}^{(T)})}{\Pr(\mathbf{X}^{(-t)} = \mathbf{x}^{(-t)})} = \frac{\boldsymbol{\alpha}_{t-1}\boldsymbol{\Gamma}\mathbf{P}(x)\boldsymbol{\beta}_t'}{\boldsymbol{\alpha}_{t-1}\boldsymbol{\Gamma}\boldsymbol{\beta}_t'}.$$

Comparing the conditional distribution for an observation with the actual observed value can reveal potential outliers in the dataset. A potential outlier is an observation which seems improbable given the remaining data, i.e. when $\Pr(X_t = x | \mathbf{X}^{(-t)} = \mathbf{x}^{(-t)})$ is small. See also Figure 5.1 in Zucchini09.

R code for calculating conditional distributions is found in appendix A.2.9 in Zucchini09.

Forecast distribution

Forecasting in an HMM is to calculate the probability distribution of the observation x at some future time $T+h$, where h is a positive integer called the forecast horizon.

Observation forecast is then:

$$\begin{aligned}\Pr(X_{T+h} = x | \mathbf{X}^{(T)} = \mathbf{x}^{(T)}) &= \frac{\Pr(\mathbf{X}^{(T)} = \mathbf{x}^{(T)}, X_{T+h} = x)}{\Pr(\mathbf{X}^{(T)} = \mathbf{x}^{(T)})} \\ &= \frac{\boldsymbol{\alpha}_T \boldsymbol{\Gamma}^h \mathbf{P}(x) \mathbf{1}'}{\boldsymbol{\alpha}_T \mathbf{1}'} \\ &= \boldsymbol{\phi}_T \boldsymbol{\Gamma}^h \mathbf{P}(x) \mathbf{1}',\end{aligned}$$

where $\boldsymbol{\phi}_T = \boldsymbol{\alpha}_T / (\boldsymbol{\alpha}_T \mathbf{1}')$.

With the forecast distribution it is easy to report: mean, median, mode, intervals etc (see table 5.1 in Zucchini09).

R code for calculating forecast distributions is found in appendix A.2.8 in Zucchini09.

State prediction

State prediction in an HMM is to calculate the probability distribution of the state i at some future time $T+h$.

State prediction:

$$\begin{aligned}\Pr(C_{T+h}=i|\mathbf{X}^{(T)}=\mathbf{x}^{(T)}) &= \frac{\Pr(C_{T+h}=i,\mathbf{X}^{(T)}=\mathbf{x}^{(T)})}{\Pr(\mathbf{X}^{(T)}=\mathbf{x}^{(T)})}\\ &= \frac{\boldsymbol{\alpha}_T\boldsymbol{\Gamma}^h(,i)}{\boldsymbol{\alpha}_T\mathbf{1}'}\\ &= \boldsymbol{\phi}_T\boldsymbol{\Gamma}^h(,i),\end{aligned}$$

where $\boldsymbol{\Gamma}^h(,i)$ denotes the i'th column of the matrix $\boldsymbol{\Gamma}^h$, which contains the probabilities of jumping into state i from all states. Note that as $h\to\infty$, $\Pr(C_{T+h}=i|\mathbf{X}^{(T)}=\mathbf{x}^{(T)})$ approaches the stationary distribution of the Markov chain.

R code for calculating state predictions is found in appendix A.2.7 in Zucchini09.

Local decoding

Decoding of an HMM refers to the procedure of determining the hidden states that most likely gave rise to the observed data. Local decoding does this "locally" in time, i.e. the most likely states are found at each time step separately.

Consider the smoothed distribution

$$\Pr(C_t = i | \mathbf{X}^{(T)} = \mathbf{x}^{(T)}) = \frac{\Pr(C_t = i, \mathbf{X}^{(T)} = \mathbf{x}^{(T)})}{\Pr(\mathbf{X}^{(T)} = \mathbf{x}^{(T)})} = \frac{\alpha_t(i)\beta_t(i)}{L_T}.$$

For each $t \in \{1, \ldots, T\}$ the most probable state i_t^* is then

$$i_t^* = \arg\max_i \Pr(C_t = i | \mathbf{X}^{(T)} = \mathbf{x}^{(T)}),$$

i.e. the mode of the smoothed distribution at time t. This maximization is called local decoding.

Important: Using local decoding to construct a sequence of most probable states can result impossible sequences since state transitions are not taken into account. That is, there is no guarantee that $\Pr(C_{t+1} = i_{t+1}^* | C_t = i_t^*) > 0$.

R code for local decoding is found in appendix A.2.5 and A.2.6 in Zucchini09.

Local decoding

Alternative decodings

One could easily think of other ways to summarize the smoothed distribution $\Pr(C_t = i|\mathbf{X}^{(T)} = \mathbf{x}^{(T)})$ than using the mode as presented on the previous slide.

For example the conditional expectation of a function g of the state C_t could be used (local expectation):

$$\begin{aligned}\widehat{g}_t &= \mathrm{E}[g(C_t)|\mathbf{X}^{(T)} = \mathbf{x}^{(T)}] \\ &= \sum_{i=1}^{m} g(i)\Pr(C_t = i|\mathbf{X}^{(T)} = \mathbf{x}^{(T)}).\end{aligned}$$

For a Poisson-HMM we might simply define $g(i) = \lambda_i$. Then the conditional expectation is a sum of the state parameters weighted according to the smoothed distribution. Note that this will most likely lead to the expectation not coinciding with a specific state, however in most cases this will not matter. An exception is when $\Pr(C_t = i|\mathbf{X}^{(T)} = \mathbf{x}^{(T)})$ is multimodal. Then the conditional expectation is inappropriate (why?).

In a similar fashion we could also compute the median, intervals etc. of $\Pr(C_t = i|\mathbf{X}^{(T)} = \mathbf{x}^{(T)})$.

Global decoding

Global decoding is the procedure of finding the state sequence $\mathbf{c}^{(T)} = (c_1, c_2, \ldots, c_T)$ which maximizes the probability $\Pr(\mathbf{C}^{(T)} = \mathbf{c}^{(T)}, \mathbf{X}^{(T)} = \mathbf{x}^{(T)})$. That is

$$\widehat{\mathbf{c}}^{(T)} = \arg\max_{\mathbf{c}^{(T)}} \Pr(\mathbf{C}^{(T)} = \mathbf{c}^{(T)}, \mathbf{X}^{(T)} = \mathbf{x}^{(T)}).$$

This is very different to local decoding which looked at state separately. Thus, the sequence $\widehat{\mathbf{c}}^{(T)}$ is likely to be different from the sequence of locally decoded states $(i_1^*, \ldots, i_T^*)$.

A brute force approach to the maximization would involve m^T function evaluations which is infeasible in general. Instead use the Viterbi algorithm (Viterbi, 2006) as described in the following.

Global decoding

Viterbi algorithm

Now we explain the Viterbi algorithm for doing global decoding of the HMM:

First define

$$\xi_{1i} = \Pr(C_1 = i, X_1 = x_1) = \delta_i p_i(x_1)$$

and, for $t = 2, 3, \ldots, T$,

$$\xi_{ti} = \max_{\mathbf{c}^{(t-1)}} \Pr(\mathbf{C}^{(t-1)} = \mathbf{c}^{(t-1)}, C_t = i, \mathbf{X}^{(T)} = \mathbf{x}^{(T)}).$$

In Viterbi's terminology ξ_{ti} is called the "state metric".

The state metric can be interpreted as follows: Of all the state sequences $(\mathbf{C}^{(t-1)} = \mathbf{c}^{(t-1)})$ leading to i by time t, ξ_{ti} is the probability of the most probable conditional on the observed data.

Global decoding

Viterbi algorithm

For $t = 2, 3, \ldots, T$, $i = 1, 2, \ldots, m$ we have the following recursion

$$\xi_{tj} = \{\max_i(\xi_{t-1,i}\,\gamma_{ij})\}p_j(x_t).$$

Check for $t = 2$:

$$\begin{aligned}
\xi_{2j} &= \max_i \Pr(C_1 = i, C_2 = j, X_1 = x_1, X_2 = x_2) \\
&= \max_i \Pr(C_2 = j, X_2 = x_2 | C_1 = i, X_1 = x_1) \Pr(C_1 = i, X_1 = x_1) \\
&= \max_i \Pr(C_2 = j, X_2 = x_2 | C_1 = i)\xi_{1i} \\
&= \max_i \Pr(X_2 = x_2 | C_2 = j, C_1 = i) \Pr(C_2 = j | C_1 = i)\xi_{1i} \\
&= \{\max_i \xi_{1i} \Pr(C_2 = j | C_1 = i)\} \Pr(X_2 = x_2 | C_2 = j) \\
&= \{\max_i(\xi_{1i}\,\gamma_{ij})\}p_j(x_2).
\end{aligned}$$

This can analogously be extended to all t (exercise 1 in Zucchini09).

Global decoding

The maximizing state sequence $\widehat{\mathbf{c}}^{(T)} = (i_1, \ldots, i_T)$ can be found with a recursion in reverse time

$$i_T = \arg \max_{i=1,\ldots,m} \xi_{Ti}$$

and, for $t = T-1, T-2, \ldots, 1$, from

$$i_t = \arg \max_{i=1,\ldots,m} (\xi_{ti} \gamma_{i,i_{t+1}}).$$

To avoid numerical underflow it is common to apply the logarithm in the maximizations. R code is found in appendix A.2.4 in Zucchini09.

As an alternative to the recursion above we can store the sequence of states during the recursion for ξ_{tj}. This requires more memory, but less CPU time. When storing the sequence Tm^2 function evaluations and Tm storage are required as opposed to $T(m^2+m)$ function evaluations when using the backward recursion.

As sequence estimate, global decoding is preferred since state transition probabilities between succeeding states ARE taken into account (in contrast to local decoding). Global decoding can however be slow for long datasets and large m.

Exercises

1,4,6

Note that there is an error in pois.HMM.local_decoding in A2.txt in Zucchini09. Here is a function that works:

```
pois.HMM.local_decoding <-
 function(x,m,lambda,gamma,delta=NULL,...)
{
 stateprobs <-
    pois.HMM.state_probs(x,m,lambda,gamma,delta=delta)
 n   <- length(x)
 ild <- rep(NA,n)
 for (i in 1:n) ild[i]<-which.max(stateprobs[,i])
 ild
}
```

References

Viterbi, A. J. 2006. A personal history of the Viterbi Algorithm. IEEE Signal Processing Magazine **23**:120–142.