Smoothing discrete data (II)
– using a hidden Markov model as implemented in the \texttt{mhsmm} package

Søren Højsgaard and Jared O’Connell

January 15, 2017

We apply the \texttt{mhsmm} package for a simple smoothing task. The data pertain to classification of a cow’s eating behaviour over time. The true eating status $S_t$ is in the vector \texttt{cls0} where ‘1’ denotes not eating and ‘2’ denotes eating. The time resolution is one minute. The observed variables $x_t$ in \texttt{cls1} are actually not observations per se but the result of a classification obtained by a neural network (using \texttt{nnet()} from the \texttt{nnet} package).

See also the vignette “Smoothing discrete data (I)” for an alternative approach to smoothing the data.

\begin{verbatim}
> load("clsX.RData")
> length(cls0)
[1] 8640
> length(cls1)
[1] 8640
> library(mhsmm)
> plot(cls1[1:2000], type='l', ylim=c(.8,2))
> addStates(cls0[1:2000])
\end{verbatim}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{plot.png}
\caption{Observed and true eating states}
\end{figure}

A simple ‘smoothing’ of the observed states can be obtained as follows:

The density function for the emission distribution is
An initial setting of the parameters is as follows:

```r
> J <- 2
> init <- c(.5,.5)
> P <- matrix(c(.9,.1,.1,.9),nrow=J)
> B <- list(pmf=matrix(.1,ncol=J,nrow=J))
> diag(B$pmf) <- .9
> init.spec <- hmspec(init,trans=P,parms.emission=B,dens.emission=dpmf)
> init.spec
```

Hidden Markov Model specification:
J (number of states): 2
init:
[1] 0.5 0.5
transition:
[,1] [,2]
[1,] 0.9 0.1
[2,] 0.1 0.9
emission:
$pmf
[,1] [,2]
[1,] 0.9 0.1
[2,] 0.1 0.9

To fit the model we need to provide the function for the M–step of the EM–algorithm:

```r
> mstep.pmf <- function(x,wt) {
+   ans <- matrix(ncol=ncol(wt),nrow=ncol(wt))
+   for(i in 1:ncol(wt))
+     for(j in 1:ncol(wt))
+       ans[i,j] <- sum(wt[which(x==j),i])/sum(wt[,i])
+   list(pmf=ans)
+ }
```

For training the model we use the first 1000 cases

```r
> samp <- 1:2640
> train <- list(s=cls0[samp], x=cls1[samp], N=length(cls0[samp]))
> valid <- list(x=cls1[-samp], N=length(cls1[-samp]))
```

We fit the model with

```r
> hmm.obj <- hmmfit(train, init.spec,mstep=mstep.pmf)
> summary(hmm.obj)
```

init:
1 0

transition:
[,1] [,2]
[1,] 0.983 0.017
[2,] 0.177 0.823

emission:
$pmf
[,1] [,2]
[1,] 0.999756954 0.0002430464
[2,] 0.008608553 0.9913914471

Two types of predictions can be made: Default is to use the Viterbi algorithm for producing the jointly most likely sequence of states given the observed data:

```r
> vit <- predict(hmm.obj, valid)
```

Alternatively we can get the individually most likely state sequence as:
> smo <- predict(hmm.obj, valid, method="smoothed")

The prediction results are quite similar:

> normtab <- function(tt) round(sweep(tt,1,rowSums(tt),"/"),2)
> SS <- cls0[-samp]
> XX <- cls1[-samp]
> cls2.vit <- vit$s
> cls2.smo <- smo$s
> normtab(table(SS,XX))

<table>
<thead>
<tr>
<th>SS</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.98</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>0.73</td>
</tr>
</tbody>
</table>

> normtab(table(SS,cls2.vit))

<table>
<thead>
<tr>
<th>cls2.vit</th>
<th>SS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>0.74</td>
</tr>
</tbody>
</table>

> normtab(table(SS,cls2.smo))

<table>
<thead>
<tr>
<th>cls2.smo</th>
<th>SS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0.98</td>
</tr>
<tr>
<td></td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>0.26</td>
</tr>
<tr>
<td></td>
<td>0.74</td>
</tr>
</tbody>
</table>

> show <- 1:2000
> cls0b <- cls0[-samp]
> cls1b <- cls1[-samp]
> c(length(SS),length(cls2.vit),length(cls2.smo), length(cls0b), length(cls1b))

[1] 6000 6000 6000 6000 6000

> plot(cls1b[show], type='l', ylim=c(.8,2))
> addStates(list(cls0b[show],cls2.vit[show], cls2.smo[show]))

![Figure 2: Observed and true eating states](image)

Figure 2: Observed and true eating states