Advanced Statistical Computing
Week 5: EM Algorithm

Aad van der Vaart

Fall 2012

Contents

EM Algorithm

EM-algorithm

SETTING:
Observation X, likelihood $\theta \mapsto p_\theta(X)$, hard to maximize and find MLE $\hat{\theta}$).

X can be viewed as 1st coordinate of (X, Y) with density $(x, y) \mapsto \bar{p}_\theta(x, y)$:

$$p_\theta(x) = \int \bar{p}_\theta(x, y)\, d\mu(y).$$

EM-ALGORITHM: GIVEN $\tilde{\theta}_0$ REPEAT

- E-step: compute $\theta \mapsto \mathrm{E}_{\tilde{\theta}_i}\big(\log \bar{p}_\theta(X, Y) | X\big)$.
- M-step: $\tilde{\theta}_{i+1} =:$ point of maximum of this function.

$\tilde{\theta}_0, \tilde{\theta}_1, \ldots$ often tends to MLE, but may not converge, converge slowly, or converge to local maximum.

[Y may be *missing data*, of *augmented data*, invented for convenience.]

EM-Algorithm — increases target

LEMMA $\tilde{\theta}_0, \tilde{\theta}_1, \ldots$ generated by EM-algorithm satisfies $p_{\tilde{\theta}_0}(X) \le p_{\tilde{\theta}_1}(X) \le \cdots$.

PROOF

$$\bar{p}_\theta(x, y) = p_\theta(y|\, x) p_\theta(x).$$

$$\mathrm{E}_{\tilde{\theta}_i}\big(\log \bar{p}_\theta(X, Y)|\, X\big) = \mathrm{E}_{\tilde{\theta}_i}\big(\log p_\theta(Y|\, X)|\, X\big) + \log p_\theta(X).$$

Because $\tilde{\theta}_{i+1}$ maximizes left side over θ, it suffices to show

$$\mathrm{E}_{\tilde{\theta}_i}\big(\log p_{\tilde{\theta}_{i+1}}(Y|\, X)|\, X\big) \le \mathrm{E}_{\tilde{\theta}_i}\big(\log p_{\tilde{\theta}_i}(Y|\, X)|\, X\big).$$

Or $-K(p, q) := \mathrm{E}_p \log(q/p)(Y) \le 0$ for $p = p_{\tilde{\theta}_i}$, $q = p_{\tilde{\theta}_{i+1}}$, conditioned on X.

Now *Kullback-Leibler divergence* $K(p; q)$ is nonnegative for any p, q.

This does not prove that $\tilde{\theta}_i$ converges to the MLE!

EM-Algorithm — linear convergence

The speed of the EM-algorithm is linear, with slow convergence if the augmented model is statistically much more informative than the data model.

Mixtures

SETTING
Observations random sample $X_1, \ldots, X_n$ from density

$$p_\theta(x) = \sum_{j=1}^{k} p_j f(x; \eta_j), \qquad \theta = (p_1, \ldots, p_k, \eta_1, \ldots, \eta_k).$$

AUGMENTED DATA

$$\mathrm{P}(Y_i = j) = p_j, \qquad X_i | \, Y_i = j \sim f(\cdot; \eta_j), \qquad i = 1, \ldots, n.$$

Full likelihood

$$p_\theta(X_1, \ldots, X_n, Y_1, \ldots, Y_n) = \prod_{i=1}^{n} \prod_{j=1}^{k} \big(p_j f(X_i; \eta_j)\big)^{1_{\{Y_i = j\}}}.$$

Mixtures — E-step, M-step

E-step: given $(\tilde{p}, \tilde{e}ta)$:

$$\mathrm{E}_{\tilde{p},\tilde{\eta}}\Big(\log \prod_{i=1}^{n}\prod_{j=1}^{k}\big(p_j f(X_i,\eta_j)\big)^{1_{\{Y_i=j\}}} \,|\, X_1,\ldots,X_n\Big)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{k} \log\big(p_j f(X_i,\eta_j)\big)\tilde{\alpha}_{i,j},$$

$$\boxed{\tilde{\alpha}_{i,j} := \mathrm{P}_{\tilde{p},\tilde{\eta}}\big(Y_i = j \,|\, X_i\big) = \frac{\tilde{p}_j f(X_i,\tilde{\eta}_j)}{\sum_c \tilde{p}_c f(X_i,\tilde{\eta}_c)}}$$

$$= \Big[\sum_{j=1}^{k} \log p_j \Big(\sum_{i=1}^{n}\tilde{\alpha}_{i,j}\Big)\Big] + \sum_{j=1}^{k}\Big[\sum_{i=1}^{n} \log f(X_i;\eta_j)\tilde{\alpha}_{i,j}\Big].$$

M-step: for $j = 1,\ldots,k$:

$$p_j^{new} = \frac{1}{n}\sum_{i=1}^{n}\tilde{\alpha}_{i,j}, \qquad \eta_j^{new} = \operatorname*{argmax}_{\eta} \sum_{i=1}^{n} \log f(X_i;\eta)\tilde{\alpha}_{i,j}.$$

[If the $f(\cdot;\eta)$ have a common parameter, then the computation of the η_j does not separate as they do here.]

EXAMPLE
If $f(\cdot;\eta) \sim N(\eta, 1)$, then

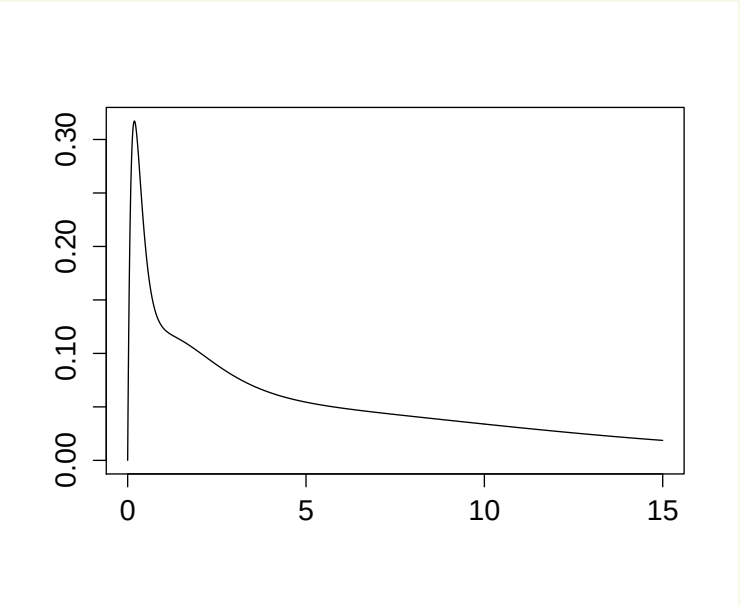
$$\sum_{i=1}^{n} \log f(X_i, \eta)\tilde{\alpha}_{i,j} = -\tfrac{1}{2}\textstyle\sum_{i=1}^{n} \tilde{\alpha}_{i,j}(X_i - \eta)^2 + Const.$$

$$\eta_j^{new} = \frac{\sum_{i=1}^{n} \tilde{\alpha}_{ij} X_i}{\sum_{i=1}^{n} \tilde{\alpha}_{i,j}}.$$

EXAMPLE
If $f(\cdot;\eta) \sim \Gamma(r, \eta)$, then

$$\sum_{i=1}^{n} \log f(X_i, \eta)\tilde{\alpha}_{i,j} = \sum_{i=1}^{n} (r \log \eta - \eta X_i)\tilde{\alpha}_{i,j} + Const.$$

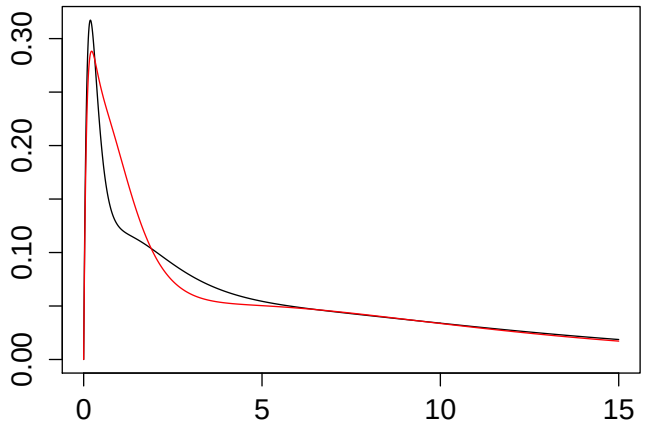
$$\eta_j^{new} = \frac{r\sum_{i=1}^{n} \tilde{\alpha}_{i,j}}{\sum_{i=1}^{n} \tilde{\alpha}_{i,j} X_i}.$$




```
> n=100
> shape=c(2,2,2); eta=c(1,6,.2); prob=c(1/4,1/8,5/8)
> component=sample(c(1,2,3),n,replace=TRUE,prob=prob)
> x=rgamma(n,shape=shape[component],rate=eta[component])
```

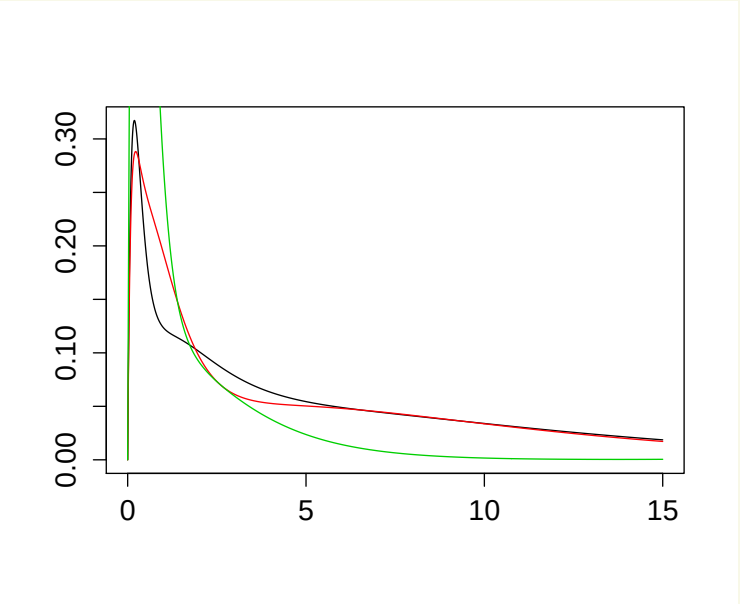
R — EM, known shape

```
> k=3; a=matrix(0,n,k); p=c(1/3,1/3,1/3); eta=c(1,2,3); change=1
> while (change>0.0001){
+    for (j in 1:k) a[,j]=p[j]*dgamma(x,2,eta[j])
+    a=diag(1/apply(a,1,sum))%*%a
+    etanew=2*apply(a,2,sum)/matrix(x,1,n)%*%a
+    pnew=apply(a,2,mean)
+    change=sum(abs(etanew-eta)+abs(pnew-p))
+    print(rbind(pnew,etanew))
+    eta=etanew; p=pnew}
[ --- output deleted ---- ]
          [,1]      [,2]      [,3]
pnew 0.6259239 0.3161804 0.05789564
     0.2157931 1.7430514 7.57683781
```



R — packages

```
> library(mixtools)
> mod=gammamixEM(x,k=3)
number of iterations= 323
> summary(mod)
Error in summary.mixEM(mod) : Unknown mixEM object of type gammamixEM
> mod[[2]]; mod[[3]]
[1] 0.37441469 0.57523322 0.05035209
          comp.1   comp.2     comp.3
alpha 1.6203475 2.092346 20.9880430
beta  0.6184701 4.126267  0.7926715
```



[Besides package `mixtools`, there is also `flexmix`, and ... (?)]

Mixtures — warnings

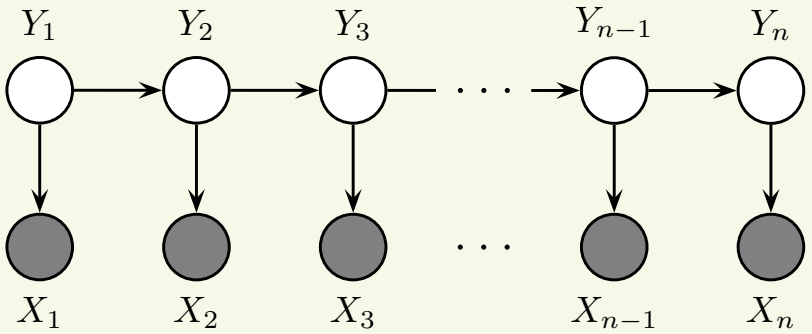
Not all mixtures are identifiable from the data: multiple parameter vectors may give the same mixture.

Maximum likelihood may work only if the parameter set is restricted. (Notable example: location scale mixtures, if the scale parameter approaches zero, the likelihood may tend to infinity.)

EM tends to be slow for large data sets, and might get stuck in local maxima (?)

Hidden Markov models

Hidden Markov model



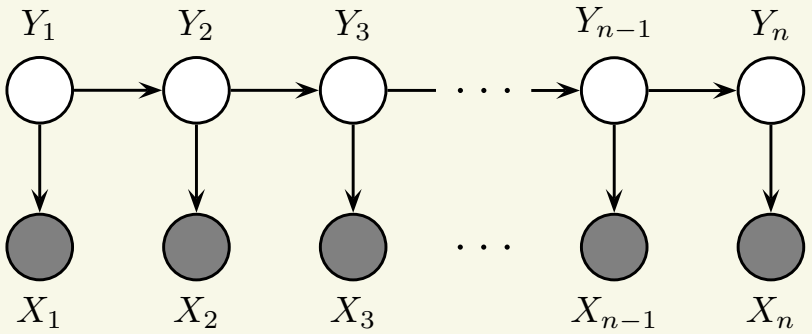
Markov chain of *hidden states* $Y_1, Y_2, \ldots,$; only *outputs* $X_1, X_2, \ldots$ observed.

X_i given Y_i conditionally independent of all other variables.

EXAMPLES

- speech recognition: states abstract, outputs Fourier coding of sounds.
- genomics: states are introns/exons, outputs nucleotides
- genomics: states are # chromosomal duplicates, outputs noisy measurements
- genetics: states inheritance vectors, output measured markers.
- cell biology: states of ion channels, outputs current or no current
- economics: state of economy, output # firms in default.

Hidden Markov model



Markov chain of *hidden states* $Y_1, Y_2, \ldots,$; only *outputs* $X_1, X_2, \ldots$ observed.

X_i given Y_i conditionally independent of all other variables.

Parameters

- density π of Y_1
- transition density $p(y_i | y_{i-1})$ of the Markov chain.
- output density $q(x_i | y_i)$.

Full likelihood

$$\pi(y_1)p(y_2 | y_1) \times \cdots \times p(y_n | y_{n-1})\ q(x_1 | y_1) \times \cdots \times q(x_n | y_n).$$

HMM — E and M-step

E-step:

$$\begin{aligned}
&\mathrm{E}_{\tilde{\pi},\tilde{p},\tilde{q}}\Big(\log\pi(Y_1)\prod_{i=2}^{n}p(Y_i|\,Y_{i-1})\prod_{i=1}^{n}q(X_i|\,Y_i)|\,X_1,\ldots,X_n\Big)\\
&\qquad=\ \mathrm{E}_{\tilde{\pi},\tilde{p},\tilde{q}}\Big(\log\pi(Y_1)|\,X_1,\ldots,X_n\Big)\\
&\qquad\quad+\sum_{i=2}^{n}\mathrm{E}_{\tilde{\pi},\tilde{p},\tilde{q}}\Big(\log p(Y_i|\,Y_{i-1})|\,X_1,\ldots,X_n\Big)\\
&\qquad\quad+\sum_{i=1}^{n}\mathrm{E}_{\tilde{\pi},\tilde{p},\tilde{q}}\Big(\log q(X_i|\,Y_i)|\,X_1,\ldots,X_n\Big).
\end{aligned}$$

M-step:

- depends on the specification of models for π, p, q.
- if state space is finite p is typically left free.
- only current estimate of law of (Y_{i-1}, Y_i) given $X_1, \ldots, X_n$ needed, which are computed using the *forward* and *backward algorithm*.

Baum-Welch

The EM-algorithm for the HMM with finite state space, and completely unspecified distributions π, p, q, is called *Baum-Welch algorithm*.

If π and p are left free:

$$\pi^{new} = p_{\tilde{\pi},\tilde{p},\tilde{q}}^{Y_1|X_1,\ldots,X_n}(y).$$

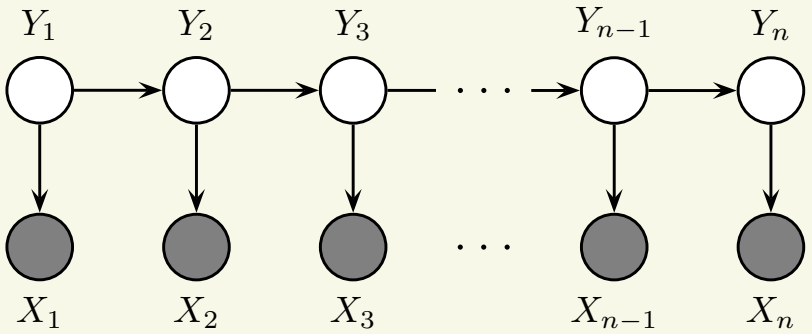
$$p^{new}(v|\,u) = \frac{\sum_{i=2}^{n} p_{\tilde{\pi},\tilde{p},\tilde{q}}^{Y_{i-1},Y_i|X_1,\ldots,X_n}(u,v)}{\sum_{i=2}^{n} p_{\tilde{\pi},\tilde{p},\tilde{q}}^{Y_{i-1}|X_1,\ldots,X_n}(u)}.$$

If q is also left free (possible for finite output space, but not often the case):

$$q^{new}(x|\,y) = \frac{\sum_{i:X_i=x} p_{\tilde{\pi},\tilde{p},\tilde{q}}^{Y_i|X_1,\ldots,X_{i-1},X_i=x,X_{i+1},\ldots,X_n}(y)}{\sum_{x\in\mathcal{X}}\sum_{i:X_i=x} p_{\tilde{\pi},\tilde{p},\tilde{q}}^{Y_i|X_1,\ldots,X_{i-1},X_i=x,X_{i+1},\ldots,X_n}(y)}.$$

[To compute these expressions need density of (Y_{i-1}, Y_i) given $X_1, \ldots, X_n$. This is computed using the *forward* and *backward algorithm*.]

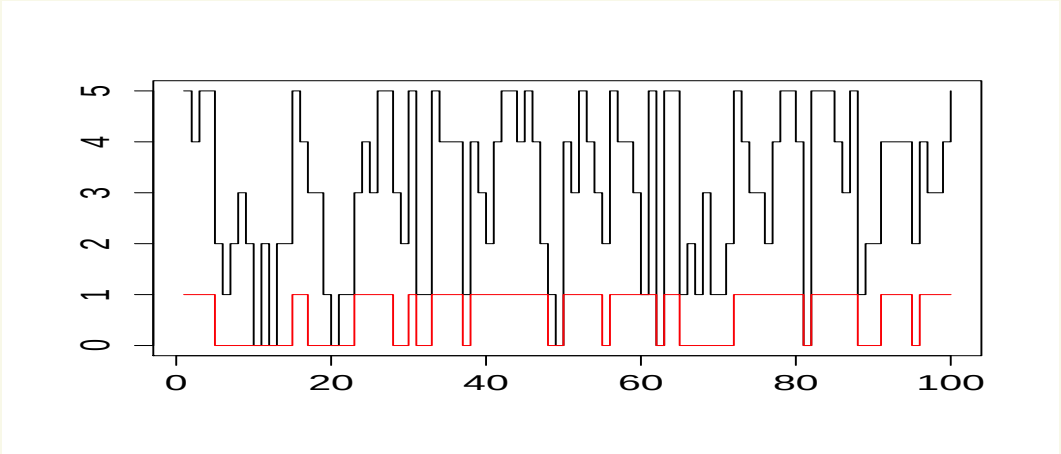
Viterbi



The Viterbi algorithm computes the most likely state path given the outcomes:

$$\underset{y_1,\ldots,y_n}{\operatorname{argmax}} \mathrm{P}(Y_1 = y_1, \ldots, Y_n = y_n \,|\, X_1, \ldots, X_n).$$

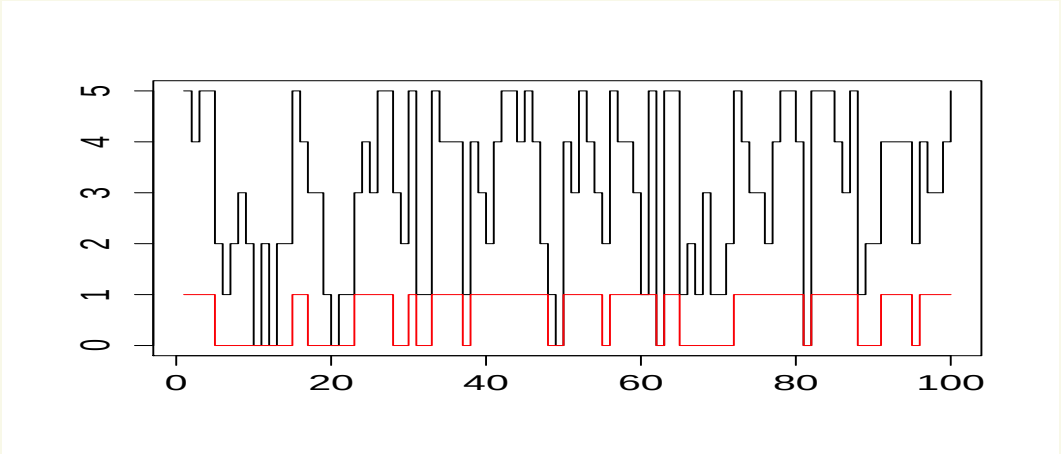
R




```
> library(HiddenMarkov)
> Pi=matrix(c(0.7,0.3,0.2,0.8),2,2,byrow=TRUE); delta=c(0.3,0.7
> n=100; pn=list(size=rep(5,n)); pm=list(prob=c(0.3,0.8))
> myhmm=dthmm(NULL,Pi=Pi,delta=delta,distn="binom",pn=pn,pm=pm)
> x=simulate(myhmm,n)
>
> plot(1:n,x$x,type="s",xlab="",ylab="")
> lines(1:n,x$y-1,col=2,type="s")
```

[Markov chain with two states, transition matrix Π, initial distribution δ. Outputs are from the binomial$(5, p)$- distribution, with $\theta = 0.3$ from state 1 and $\theta = 0.8$ from state 2. Red: states, Black: outputs.]

R

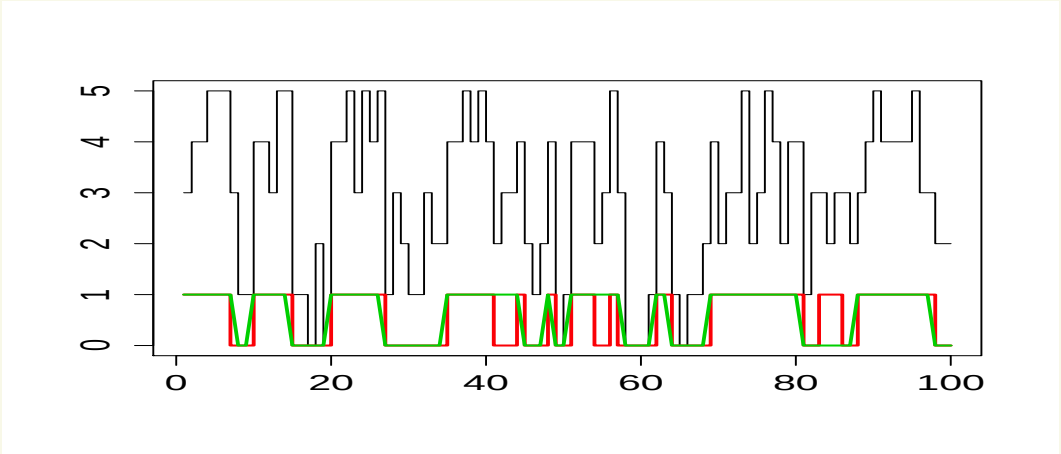



```
> mod=BaumWelch(x); mod$Pi; mod$pm
[---- output deleted ---]
          [,1]      [,2]
[1,] 0.6287149 0.3712851
[2,] 0.2637289 0.7362711
$prob
[1] 0.3173456 0.8313127
```

[Markov chain with two states, transition matrix $\Pi = \begin{matrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{matrix}$, initial distribution $\delta = (0.3, 0.7)$.

Outputs are from the binomial$(5, p)$- distribution, with $\theta = 0.3$ from state 1 and $\theta = 0.8$ from state 2.]

R




```
> Viterbi(x)
  [1] 2 2 2 2 2 2 2 1 1 2 2 2 2 2 1 1 1 1 1 2 2 2 2 2 2 2 1 1 1
 [36] 2 2 2 2 2 2 2 2 2 1 1 1 2 1 1 2 2 2 2 2 2 2 1 1 1 1 2 2 1
 [71] 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 1 1
> lines(1:n,Viterbi(x)-1,col=3,lw=2)
```

[Red: true states, Black: outputs; Green: reconstructed states.]