Lab2 oskhi827

Oskar Hidén - oskhi827

9/17/2020

Question 1

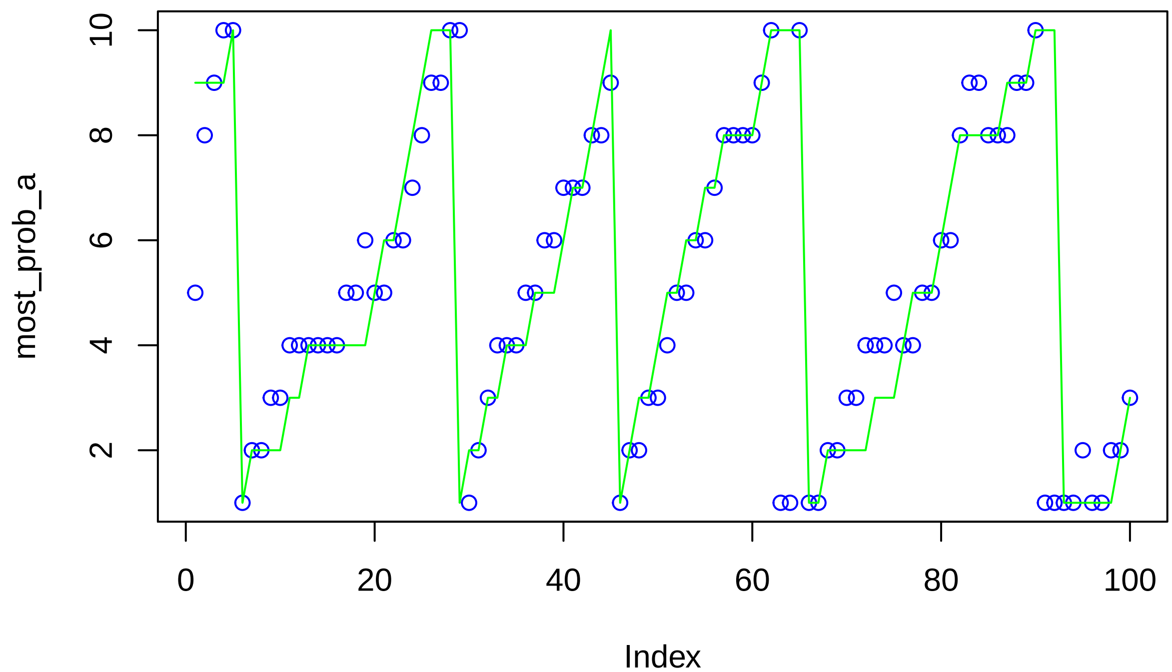
```
library(HMM)

states = c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10")
symbols = c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10")
#start_prob = rep(0, 10)
#start_prob[1] = 1
start_prob = NULL
sur_state = function(x){
  state = x%%10
  if (state ==0) {
    state=10
  }
  return(state)
}
trans_prob = matrix(data=0, nrow = 10, ncol=10)
for (i in 1:10) {
  trans_prob[i,i] = 0.5
  trans_prob[i,sur_state(i+1)] = 0.5
}
emmis_prob = matrix(data=0, nrow = 10, ncol=10)
for (i in 1:10) {
  for (j in -2:2) {
    emmis_prob[i,sur_state(i+j)] = 0.5
  }
}
HMM = initHMM(states, symbols, start_prob, trans_prob, emmis_prob)
```

Question 2

```
set.seed(12345)
N = 100
sim = simHMM(HMM, N)
```

Question 3 & 4

```r
observed = sim$observation

# filterd alpha --Alpha uses all observations up to point t to estimate Zt
alpha_log = forward(HMM, observed)
alpha = exp(alpha_log)
most_prob_a = apply(alpha, MARGIN = 2, which.max)
plot(most_prob_a, col="blue")
lines(sim$states, col="green")
```

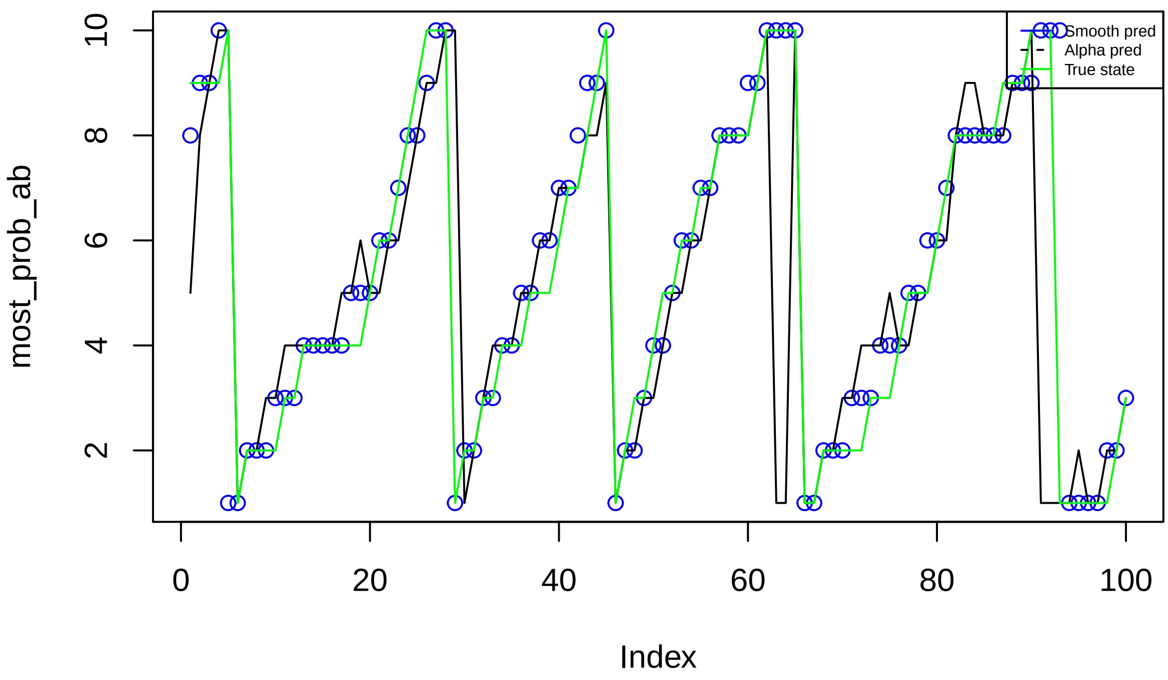



```r
fi_acc = sum(sim$states==most_prob_a)/100
cat("Filtered accuracy:", fi_acc)
```

```
## Filtered accuracy: 0.53
```

```r
# Beta.
beta_log = backward(HMM, observed)
beta = exp(beta_log)

# smoothed alpha*beta -- Alpha beta uses all observations (to T) to estimate Zt. "which is better"
alpha_beta = alpha*beta
most_prob_ab = apply(alpha_beta, MARGIN = 2, which.max)
```

```r
plot(most_prob_ab, col="blue",)
lines(most_prob_a)
lines(sim$states, col="green")
legend("topright", c("Smooth pred", "Alpha pred", "True state"),
       col=c("blue", "black", "green"), lty=1:2, cex=0.5)
```

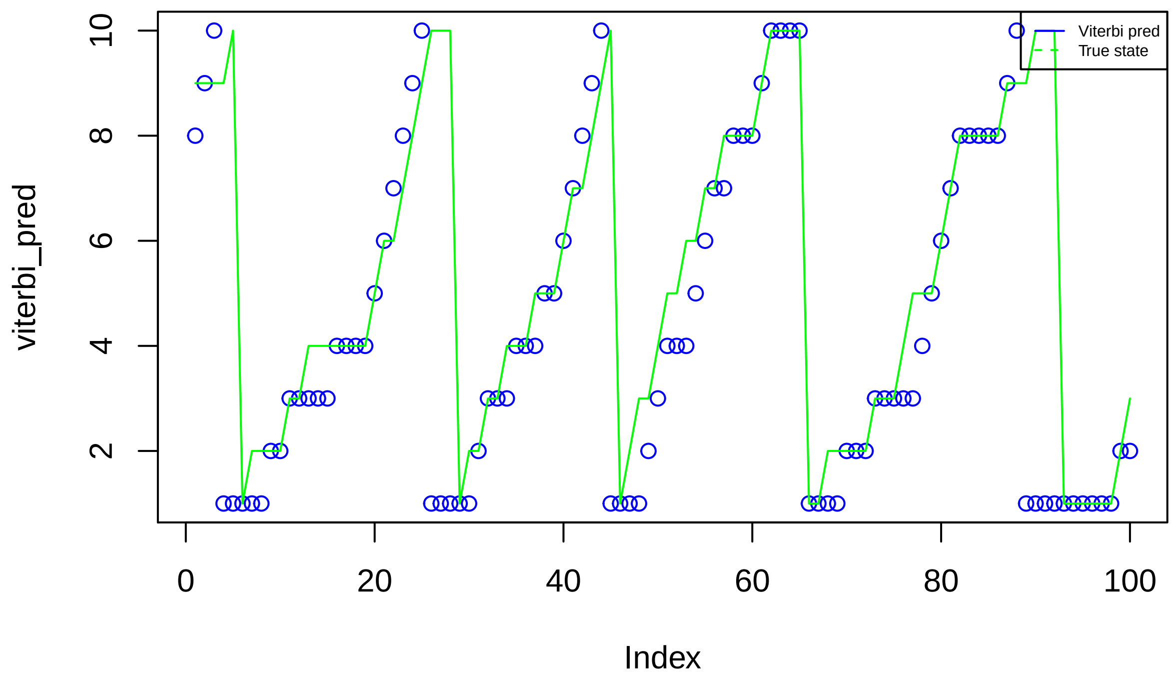



```r
sm_acc = sum(sim$states==most_prob_ab)/100
cat("Smoothing accuracy:", sm_acc)
```

```
## Smoothing accuracy: 0.74
```

```r
# Normalize
norm_fact =  apply(alpha, 2, sum)
filtered = apply(alpha, 1, "/", norm_fact )
filtered = t(filtered)

norm_fact = apply(alpha_beta, 2, sum)
smoothing = apply(alpha_beta, 1, "/", norm_fact)
smoothing = t(smoothing)

# Most prob path
viterbi_pred = viterbi(HMM, observed)
plot(viterbi_pred, col="blue",)
lines(sim$states, col="green")
```

```r
legend("topright", c("Viterbi pred", "True state"),
       col=c("blue", "green"), lty=1:2, cex=0.5)
```




```r
vi_acc = sum(sim$states==viterbi_pred)/100
cat("Most prob path accuracy:", vi_acc)
```

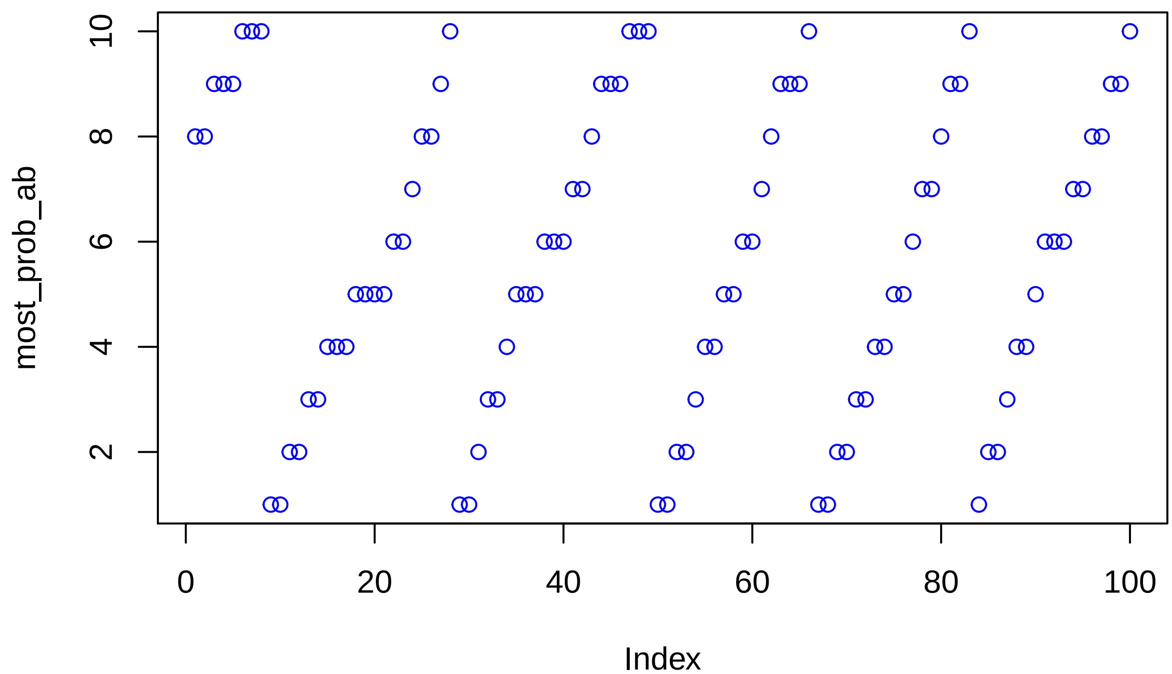
```
## Most prob path accuracy: 0.56
```

Question 5

```r
# new simulatons
sim = simHMM(HMM, N)
observed = sim$observation

# filterd alpha --Alpha uses all observations up to point t to estimate Zt
alpha = exp(forward(HMM, observed))
most_prob_a = apply(alpha, MARGIN = 2, which.max)

fi_acc = sum(sim$states==most_prob_a)/100
cat("Filtered accuracy:", fi_acc)
```

```
## Filtered accuracy: 0.46
```

```r
# Beta.
beta = exp(backward(HMM, observed))

# smoothed alpha*beta -- Alpha beta uses all observations (to T) to estimate Zt. "which is better"
alpha_beta = alpha*beta
most_prob_ab = apply(alpha_beta, MARGIN = 2, which.max)
plot(most_prob_ab, col="blue",)
```




```r
sm_acc = sum(sim$states==most_prob_ab)/100
cat("Smoothing accuracy:", sm_acc)
```

```
## Smoothing accuracy: 0.68
```

```r
# Most prob path
viterbi_pred = viterbi(HMM, observed)

vi_acc = sum(sim$states==viterbi_pred)/100
cat("Most prob path accuracy:", vi_acc)
```

```
## Most prob path accuracy: 0.61
```

The filtered prediction uses all obervations up to the time t to predict Xt, where the smoothed prediciton uses all observed values (X0. . . XT) in its prediction. Smoothed prediction uses more information and therfore it should in general be more accurate.

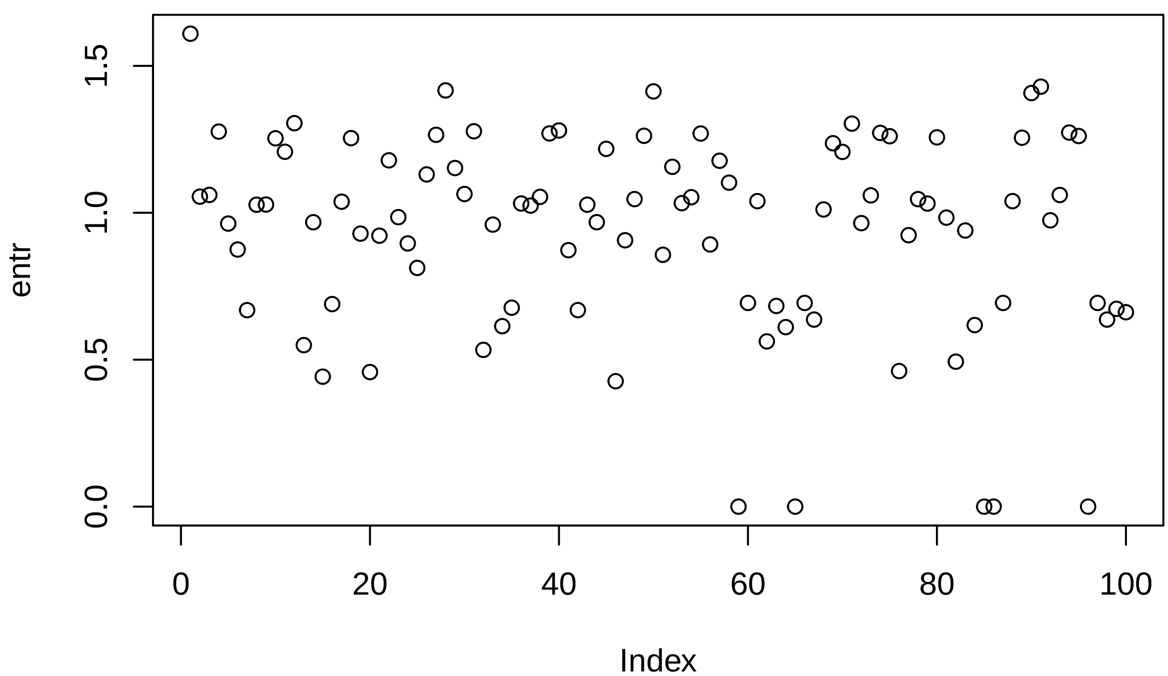
In general smoothed is also better than viterbi (most prob path), beacause viterbi has the same observations but with one extra Constrain, a valid path. Therfore the accuracy for viterbi would be less or equal to smoothed.

Question 6

```
library(entropy)
entr = apply(filtered, 2, entropy.empirical)
entropy.empirical(rep(0.1,10)) # Max entrpy value!
```

```
## [1] 2.302585
```

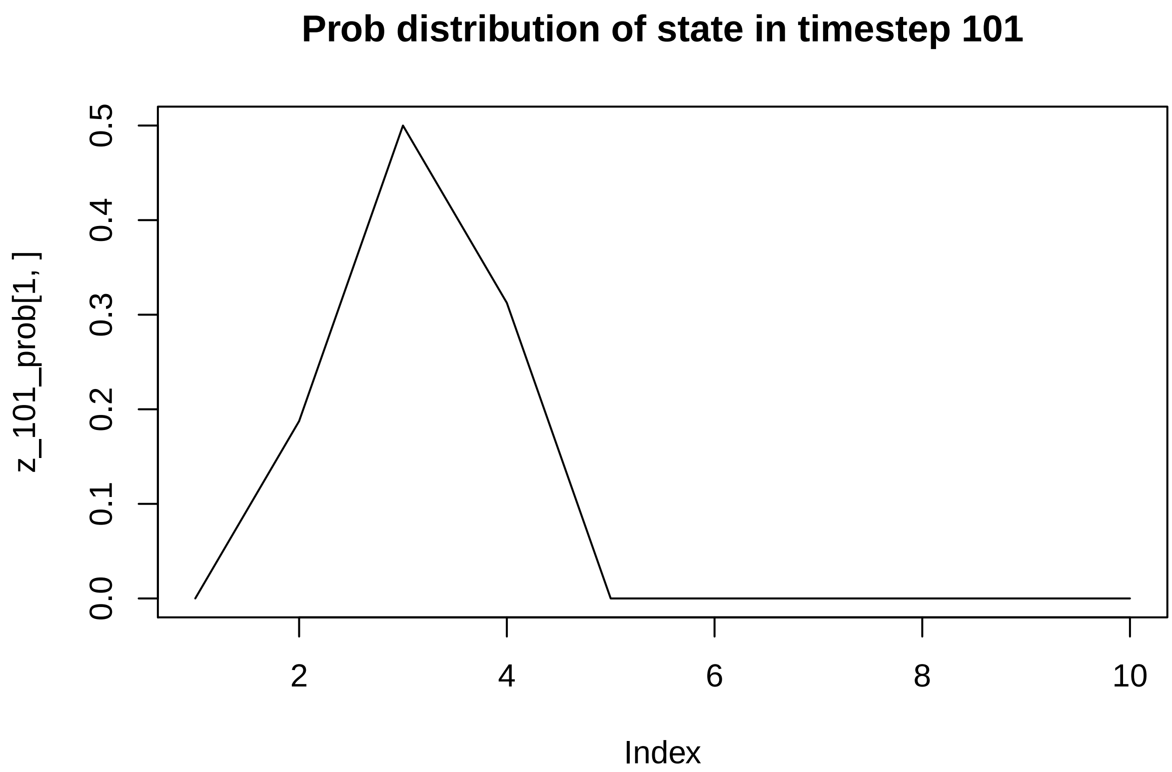
```
plot(entr)
```



In the first iterations the entrpy drops since we get more information, and therfore we get a better prediction. But after a couple of iterations the entropy increases again. Therfore we do not always get a better prediction with more observations. For some t the entropy is 0, and then we are sure of our predicton, but then the entropy increases agin. The certainty depends more on the combinations of observatins than on the total amount of data.

Question 7

```r
# Last prediction distribution of Z
z_T_prob = filtered[,dim(filtered)[2]]

z_101_prob = z_T_prob%*%HMM$transProbs
plot(z_101_prob[1,], type="l", main = "Prob distribution of state in timestep 101")
```




```r
z_101_pred = which.max(z_101_prob)
cat("Prediction of state in timestep 101 is: ", z_101_pred)
```

```
## Prediction of state in timestep 101 is:  3
```