Introduction to Lcpp

Jan-Ole Koslik

The package `Lcpp` provides convenient **R** wrapper functions for the **forward algorithm** used to fit **hidden Markov models** (HMMs), **hidden semi-Markov models** (HSMMs) and **state space models** (SSMs) via **direct numerical maximum likelihood estimation**, as well as auxiliary functions that make previous steps in the likelihood computation very fast and handy.

The three main families of functions are `forward`, `tpm` and `stationary` and we showcasse the simplest versions in the following introductory example.

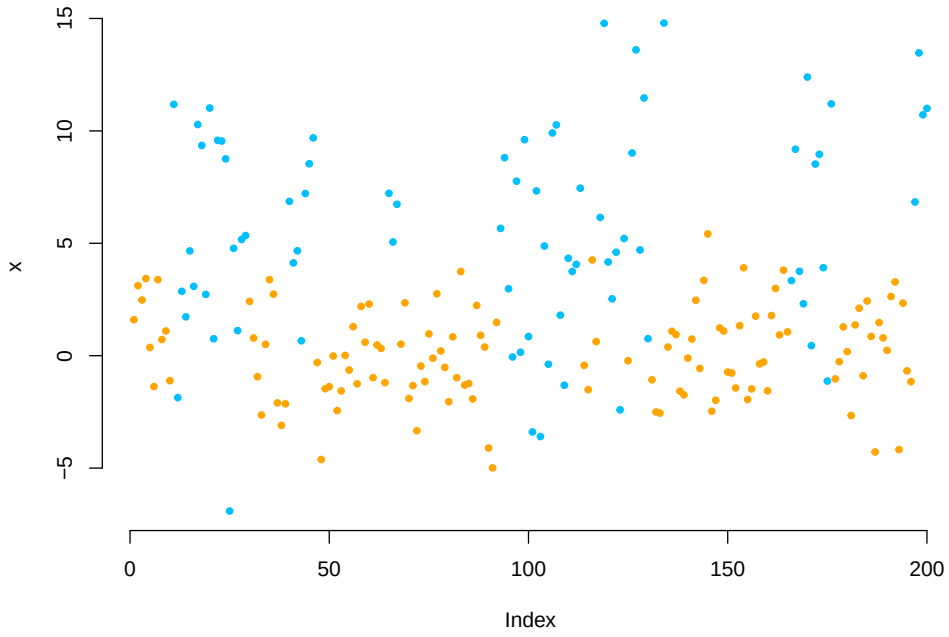
Introductory example: Homogeneous HMM

Generating data from a 2-state HMM Here we can use `stationary()` to compute the stationary distribution.

```
# parameters
mu = c(0, 6)
sigma = c(2, 4)
Gamma = matrix(c(0.95, 0.05, 0.15, 0.85), nrow = 2, byrow = TRUE)
delta = stationary(Gamma) # stationary HMM

# simulation
n = 1000
set.seed(123)
s = x = rep(NA, n)
s[1] = sample(1:2, 1, prob = delta)
x[1] = stats::rnorm(1, mu[s[1]], sigma[s[1]])
for(t in 2:n){
  s[t] = sample(1:2, 1, prob = Gamma[s[t-1],])
  x[t] = stats::rnorm(1, mu[s[t]], sigma[s[t]])
}

plot(x[1:200], bty = "n", pch = 20, ylab = "x",
     col = c("orange","deepskyblue")[s[1:200]])
```



Writing the negative log-likelihood function Within the negative log-likelihood function we build the transition probability matrix using the `tpm()` function, compute the stationary distribution using `stationary()` and calculate the log-likelihood using `forward()` in the last line.

```
mllk = function(theta.star, x){
  # parameter transformations for unconstraint optimization
  Gamma = tpm(theta.star[1:2])
  delta = stationary(Gamma) # stationary HMM
  mu = theta.star[3:4]
  sigma = exp(theta.star[5:6])
  # calculate all state-dependent probabilities
  allprobs = matrix(1, length(x), 2)
  for(j in 1:2){ allprobs[,j] = stats::dnorm(x, mu[j], sigma[j]) }
  # return negative for minimization
  -forward(delta, Gamma, allprobs)
}
```

```
theta.star = c(-1,-1,1,4,log(1),log(3))
# initial transformed parameters: not chosen too well
s = Sys.time()
mod = stats::nlm(mllk, theta.star, x = x)
Sys.time()-s
#> Time difference of 0.01668596 secs
```

Fitting an HMM to the data Really fast!

Visualizing results Again, we use `tpm()` and `stationary()` to tranform the unconstraint parameters to working parameters.

```r
# transform parameters to working
Gamma = tpm(mod$estimate[1:2])
delta = stationary(Gamma) # stationary HMM
mu = mod$estimate[3:4]
sigma = exp(mod$estimate[5:6])

hist(x, prob = TRUE, bor = "white", breaks = 40, main = "")
curve(delta[1]*dnorm(x, mu[1], sigma[1]), add = TRUE, lwd = 2, col = "orange", n=500)
curve(delta[2]*dnorm(x, mu[2], sigma[2]), add = TRUE, lwd = 2, col = "deepskyblue", n=500)
curve(delta[1]*dnorm(x, mu[1], sigma[1])+delta[2]*dnorm(x, mu[2], sigma[2]),
      add = TRUE, lwd = 2, lty = "dashed", n=500)
legend("topright", col = c("orange", "deepskyblue", "black"), lwd = 2, bty = "n",
       lty = c(1,1,2), legend = c("state 1", "state 2", "marginal"))
```