Hidden Markov Models

ed: Eli Gurarie

QERM 598 - Lecture 8b

May 23, 2013

Hidden Markov Models

Sometimes we are interested in inferring a "hidden state" X that underlies some observed process Y. These are called **Hidden Markov Models** (HMM).

Example applications:

- Speech recognition: States are *words*, observations are *sounds*, transition probability structure is *language*.
- Climate: what regime was a climate in, based on proxy data.
- Finance/Economics: variables changing behavior due to some underlying, unobserved cause.
- Animal movement: What behavioral mode is an animal in, when only movements are observed?

Hidden Markov Models

Sometimes we are interested in inferring a "hidden state" X that underlies some observed process Y. These are called **Hidden Markov Models** (HMM).

Example applications:

- Speech recognition: States are *words*, observations are *sounds*, transition probability structure is *language*.
- Climate: what regime was a climate in, based on proxy data.
- Finance/Economics: variables changing behavior due to some underlying, unobserved cause.
- Animal movement: What behavioral mode is an animal in, when only movements are observed?

Hidden Markov Models

Sometimes we are interested in inferring a "hidden state" X that underlies some observed process Y. These are called **Hidden Markov Models** (HMM).

Example applications:

- Speech recognition: States are *words*, observations are *sounds*, transition probability structure is *language*.
- Climate: what regime was a climate in, based on proxy data.
- Finance/Economics: variables changing behavior due to some underlying, unobserved cause.
- Animal movement: What behavioral mode is an animal in, when only movements are observed?

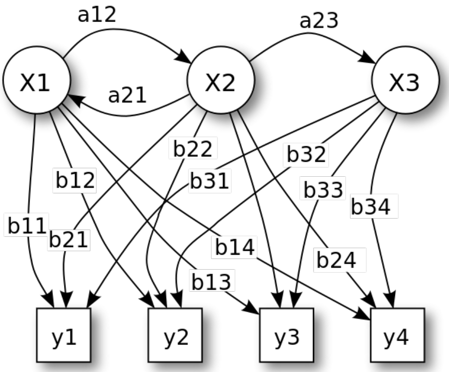
Hidden Markov Models

Sometimes we are interested in inferring a "hidden state" X that underlies some observed process Y. These are called **Hidden Markov Models** (HMM).

Example applications:

- Speech recognition: States are *words*, observations are *sounds*, transition probability structure is *language*.
- Climate: what regime was a climate in, based on proxy data.
- Finance/Economics: variables changing behavior due to some underlying, unobserved cause.
- Animal movement: What behavioral mode is an animal in, when only movements are observed?

Hidden Markov Models: Schematic



x - states
y - possible observations

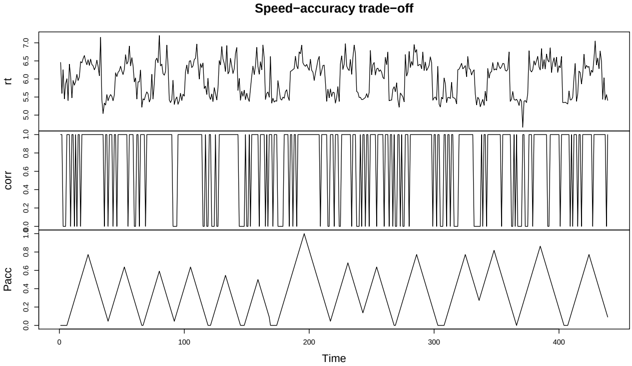
a - state transition probabilities
b - output probabilities

There are obviously lots of pieces here, and it is not generally easy to estimate an HMM!

A new-ish R package to the rescue

`depmixS4` - stands for (I think) "dependent mixture" models.
Here is data from an experiment on cognitive responses of students:[1]

```r
require(depmixS4); data(speed); plot(ts(speed), main = "Speed-accuracy trade-off")
```



The basic question is: are there different "cognitive states" that the brain switches between when facing the challenge? What are the transitions between those states? Are those transitions dependent on the reward?"

[1] Gilles Dutilh, et al. Cognitive Science (2001), 35:211-250.

From the `depmixS4` vignette:

The data have the general form $O_{1:T} = (O_1^1, \ldots, O_1^m,\ O_2^1, \ldots, O_2^m,\ \ldots,\ O_T^1, \ldots, O_T^m)$ for an m-variate time series of length T.

Joint likelihood of observations $O_{1:T}$ and latent states $S_{1:T} = (S_1, \ldots, S_T)$, given model parameters θ and covariates $z_{1:T} = (z_1, \ldots, z_T)$, can be written as:

$$(O_{1:T}, S_{1:T}|\theta, z_{1:T}) = \pi_i(z_1) b_{S_1}(O_1|z_1) \prod_{t=1}^{T-1} a_{ij}(z_t) b_{S_t}(O_{t+1}|z_{t+1}), \tag{1}$$

where:

1. S_t is an element of $\mathcal{S} = \{1 \ldots n\}$, a set of n latent classes or states.
2. $\pi_i(z_1) = (S_1 = i|z_1)$, giving the probability of class/state i at time $t = 1$ with covariate z_1.
3. $a_{ij}(z_t) = (S_{t+1} = j|S_t = i, z_t)$, provides the probability of a transition from state i to state j with covariate z_t.
4. b_{S_t} is a vector of observation densities $b_j^k(z_t) = (O_t^k|S_t = j, z_t)$ that provide the conditional densities of observations O_t^k associated with latent class/state j and covariate z_t, $j = 1, \ldots, n$, $k = 1, \ldots, m$.

For the example data above, b_j^k can be Gaussian for response-time, and a Bernoulli for accuracy.

From the `depmixS4` vignette:

The data have the general form $O_{1:T} = (O_1^1, \dots, O_1^m, O_2^1, \dots, O_2^m, \dots, O_T^1, \dots, O_T^m)$ for an m-variate time series of length T.
Joint likelihood of observations $O_{1:T}$ and latent states $S_{1:T} = (S_1, \dots, S_T)$, given model parameters θ and covariates $z_{1:T} = (z_1, \dots, z_T)$, can be written as:

$$(O_{1:T}, S_{1:T}|\theta, z_{1:T}) = \pi_i(z_1) b_{S_1}(O_1|z_1) \prod_{t=1}^{T-1} a_{ij}(z_t) b_{S_t}(O_{t+1}|z_{t+1}), \tag{1}$$

where:

1. S_t is an element of $\mathcal{S} = \{1 \dots n\}$, a set of n latent classes or states.
2. $\pi_i(z_1) = (S_1 = i|z_1)$, giving the probability of class/state i at time $t = 1$ with covariate z_1.
3. $a_{ij}(z_t) = (S_{t+1} = j|S_t = i, z_t)$, provides the probability of a transition from state i to state j with covariate z_t.
4. b_{S_t} is a vector of observation densities $b_j^k(z_t) = (O_t^k|S_t = j, z_t)$ that provide the conditional densities of observations O_t^k associated with latent class/state j and covariate z_t, $j = 1, \dots, n$, $k = 1, \dots, m$.

For the example data above, b_j^k can be Gaussian for response-time, and a Bernoulli for accuracy.

Gaussian model with two states, no covariates:

```
model <- depmix(response = rt ~ 1, data = speed, nstates = 2)
fit <- fit(model, verbose = FALSE)

## iteration 36 logLik: -88.73

summary(fit)

## Initial state probabilties model
## Model of type multinomial (identity), formula: ~1
## <environment: 0x0000000009540260>
## Coefficients:
##          [,1] [,2]
## [1,] 1.169e-61    1
##
## Transition model for state (component) 1
## Model of type multinomial (identity), formula: ~1
## <environment: 0x00000000097c83a0>
## Coefficients:
## [1] 0.8835 0.1165
##
## Transition model for state (component) 2
## Model of type multinomial (identity), formula: ~1
## <environment: 0x00000000097c83a0>
## Coefficients:
## [1] 0.08433 0.91567
##
##
## Response model(s) for state 1
##
## Response model for response 1
## Model of type gaussian (identity), formula: rt ~ 1
## Coefficients:
## [1] 5.51
## sd  0.1918
```

Gaussian model with two states and Pacc as a covariate:

```
model2 <- depmix(rt ~ 1, data = speed, nstates = 2, transition = ~scale(Pacc))
fit2 <- fit(model2, verbose = FALSE)

## iteration 32 logLik: -44.2

summary(fit2, "transition")

## Transition model for state (component) 1
## Model of type multinomial (mlogit), formula: ~scale(Pacc)
## Coefficients:
##      [,1]    [,2]
## [1,]    0 -0.9518
## [2,]    0  1.3924
## Probalities at zero values of the covariates.
## 0.7215 0.2785
##
## Transition model for state (component) 2
## Model of type multinomial (mlogit), formula: ~scale(Pacc)
## Coefficients:
##      [,1]  [,2]
## [1,]    0 2.472
## [2,]    0 3.581
## Probalities at zero values of the covariates.
## 0.07788 0.9221
```

Joint Gaussian-binomial response

```
model3 <- depmix(list(rt ~ 1, corr ~ 1), data = speed, nstates = 2, family = list(gaussian(),
    multinomial("identity")), transition = ~scale(Pacc), instart = runif(2))
fit3 <- fit(model3, verbose = FALSE, emc = em.control(rand = FALSE))

## iteration 32 logLik: -255.5

summary(fit3, "transition")

## Transition model for state (component) 1
## Model of type multinomial (mlogit), formula: ~scale(Pacc)
## Coefficients:
##      [,1]   [,2]
## [1,]    0 -2.402
## [2,]    0 -3.722
## Probalities at zero values of the covariates.
## 0.917 0.083
##
## Transition model for state (component) 2
## Model of type multinomial (mlogit), formula: ~scale(Pacc)
## Coefficients:
##      [,1]    [,2]
## [1,]    0  0.9265
## [2,]    0 -1.5986
## Probalities at zero values of the covariates.
## 0.2836 0.7164
```