Model based detection of homogeneous portions in trajectories

Marie-Pierre Etienne

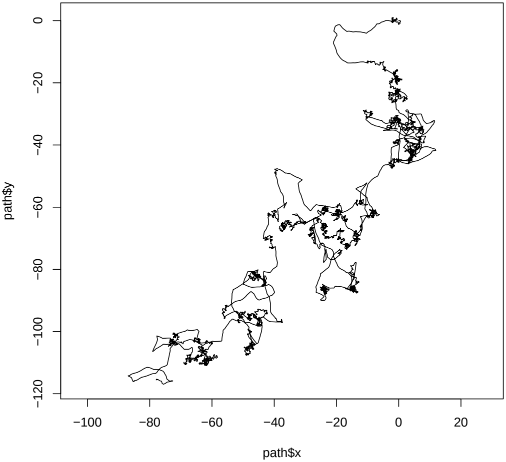
AgroParisTech / INRA

Movement Ecology Workshop 2015 - Port Elizabeth

Detection of homogenous region in trajectories

Why ?

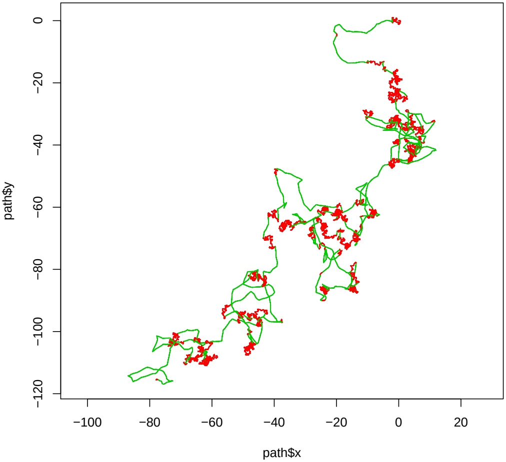
- Different behaviour
- Link with different activities
- Link with different environmental condition



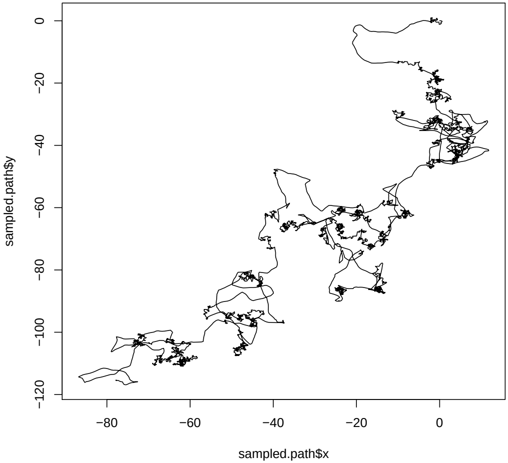
Detection of homogenous region in trajectories

Why ?

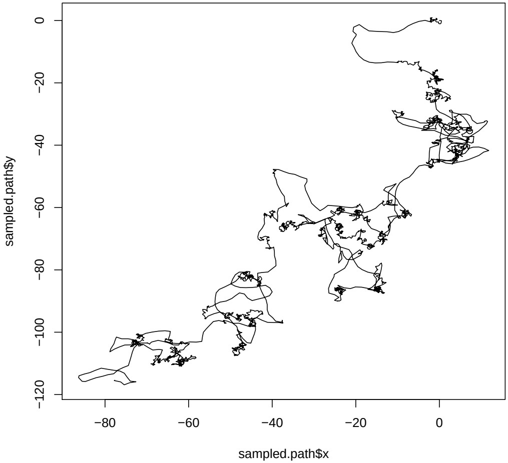
- Different behaviour
- Link with different activities
- Link with different environmental condition



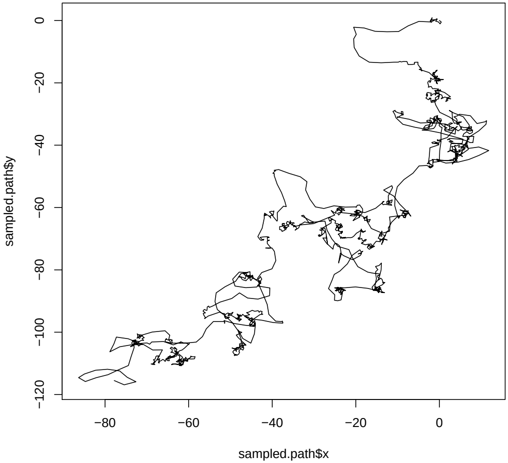
Effect of sampling step



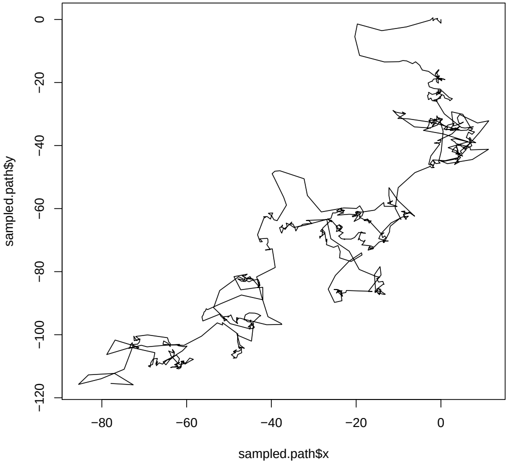
Effect of sampling step



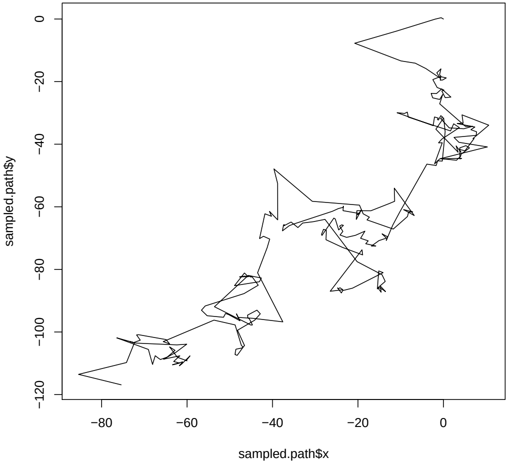
Effect of sampling step



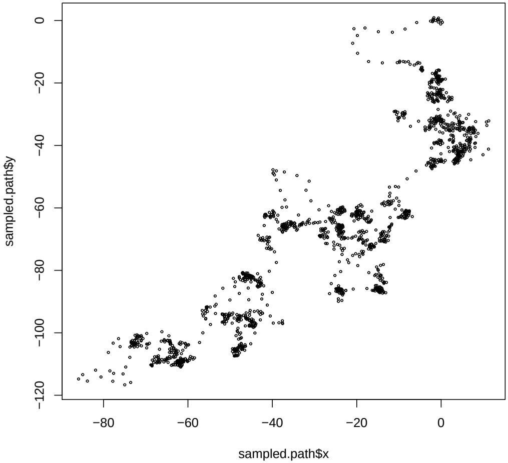
Effect of sampling step



Effect of sampling step



Effect of sampling step



Summarising trajectories

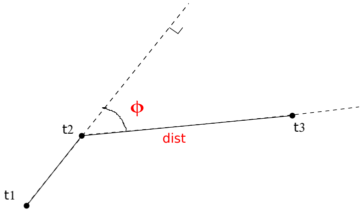
$(t_1, \ldots, t_N)$ denotes the time acquisition and $((x_1, y_1), \ldots, (x_N, y_N))$ the position at those times.

Trajetories as Turning angle and Speed

$$\mathbf{\Phi} = (\phi_2, \ldots, \phi_N)$$

$$\mathbf{S} = (S_2, \ldots, \S_N),$$

with $S_i = dist_i/(t_i - t_{i-1})$



Summarising trajectories

$(t_1, \ldots, t_N)$ denotes the time acquisition and $((x_1, y_1), \ldots, (x_N, y_N))$ the position at those times.

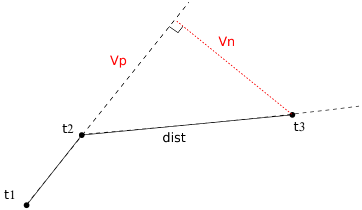
Trajetories as Persistent and Normal Velocity

$$\mathbf{V}^P = (V_2^P, \ldots, V_N^P)$$

$$\mathbf{V}^N = (V_2^N, \ldots, V_N^N)$$

with

$$V_i^P = S_i \, cos(\phi_i)$$
$$V_i^N = S_i \, sin(\phi_i)$$



From Trajectories data to signal - Loosing information on spatial location

How trajectories data migh be considered?

- A sequence of (time, position)
- Turning angle and speed sequences
- Persistent and Normal Velocity sequences

From Trajectories data to signal - Loosing information on spatial location

How trajectories data migh be considered?

- A sequence of (time, position)
- Turning angle and speed sequences
- Persistent and Normal Velocity sequences

What is affected by sampling?

- A sequence of (time, position)
- Turning angle and speed sequences
- Persistent and Normal Velocity sequences

From Trajectories data to signal - Loosing information on spatial location

How trajectories data migh be considered?

- A sequence of (time, position)
- Turning angle and speed sequences
- Persistent and Normal Velocity sequences

What is affected by sampling?

- A sequence of (time, position)
- Turning angle and speed sequences
- Persistent and Normal Velocity sequences

Model approach:

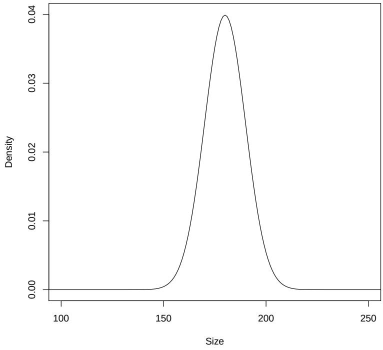
- Most methods don't consider the two phenomena : movement and sampling process.
- Results will be closely dependent of the sampling step.

Model and parameters

The model is a tentative to represent the main characteristics of potential data.

Model and parameters

The model is a tentative to represent the main characteristics of potential data.
If we observe the size of people Y and assume $Y \sim \mathcal{N}(\mu, \sigma)$



Model and parameters

The model is a tentative to represent the main characteristics of potential data.
If people are sampled at random (with no relationship)

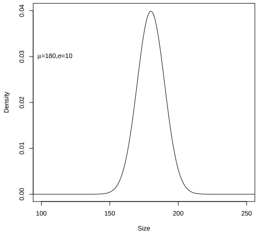
$$Y_i \overset{i.i.d}{\sim} \mathcal{N}(\mu, \sigma)$$

Model and parameters

The model is a tentative to represent the main characteristics of potential data.

$$Y_i \overset{i.i.d}{\sim} \mathcal{N}(\mu, \sigma)$$

The parameters rule the behaviour of the model.

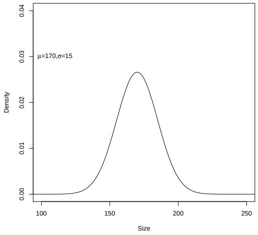


Model and parameters

The model is a tentative to represent the main characteristics of potential data.

$$Y_i \overset{i.i.d}{\sim} \mathcal{N}(\mu, \sigma)$$

The parameters rule the behaviour of the model.



Estimation

Simulation context: with a given model M, and given parameters $\boldsymbol{\theta}$, you can produce fake data.
From $(M, \boldsymbol{\theta})$ to $\mathbf{Y}$.

Estimation

Simulation context: with a given model M, and given parameters $\boldsymbol{\theta}$, you can produce fake data.
From $(M, \boldsymbol{\theta})$ to $\mathbf{Y}$.
Estimation context: with a given model M, and some data $\mathbf{Y}$, you want to determine a good value for $\boldsymbol{\theta}$.
From $(M, \mathbf{Y})$ to $\boldsymbol{\theta}$.

Estimation

Simulation context: with a given model M, and given parameters $\boldsymbol{\theta}$, you can produce fake data.
From $(M, \boldsymbol{\theta})$ to $\mathbf{Y}$.
Estimation context: with a given model M, and some data $\mathbf{Y}$, you want to determine a good value for $\boldsymbol{\theta}$.
From $(M, \mathbf{Y})$ to $\boldsymbol{\theta}$.
What is a good value for $\boldsymbol{\theta}$? with a given model M, and some data $\mathbf{Y}$, you want to give a score to each possible value of $\boldsymbol{\theta}$.

Estimation

Simulation context: with a given model M, and given parameters $\boldsymbol{\theta}$, you can produce fake data.
From $(M, \boldsymbol{\theta})$ to $\mathbf{Y}$.
Estimation context: with a given model M, and some data $\mathbf{Y}$, you want to determine a good value for $\boldsymbol{\theta}$.
From $(M, \mathbf{Y})$ to $\boldsymbol{\theta}$.
What is a good value for $\boldsymbol{\theta}$? with a given model M, and some data $\mathbf{Y}$, you want to give a score to each possible value of $\boldsymbol{\theta}$.
The likelihood as a measure of the quality of $\boldsymbol{\theta}$: with a given model M, and some data $\mathbf{Y}$, for each value of $\boldsymbol{\theta}$ you can compute the probability

$$\mathbb{P}(\mathbf{Y}; \boldsymbol{\theta})$$

Statistics and Hidden variables

A model $(M, \boldsymbol{\theta})$ produce $\mathbf{Y}$ and $\mathbf{Z}$.

Statistics and Hidden variables

A model $(M, \boldsymbol{\theta})$ produce $\mathbf{Y}$ and $\mathbf{Z}$.
The only observed data are $\mathbf{Y}$ while $\mathbf{Z}$ are hidden variables.

Statistics and Hidden variables

A model $(M, \boldsymbol{\theta})$ produce **Y** and **Z**.
The only observed data are **Y** while **Z** are hidden variables.
Questions are

- Parameters: Is it still possible to estimate $\boldsymbol{\theta}$?
- Information on **Z**: is it possible to "reconstruct" the unobserved data **Z** ?

Statistics and Hidden variables

A model $(M, \boldsymbol{\theta})$ produce $\mathbf{Y}$ and $\mathbf{Z}$.
The only observed data are $\mathbf{Y}$ while $\mathbf{Z}$ are hidden variables.
Questions are

- Parameters: Is it still possible to estimate $\boldsymbol{\theta}$?
- Information on $\mathbf{Z}$: is it possible to "reconstruct" the unobserved data $\mathbf{Z}$?

Bayes formula is the key :

$$\mathbb{P}(\mathbf{Y}, \mathbf{Z}) = \mathbb{P}(\mathbf{Y}|\mathbf{Z})\mathbb{P}(\mathbf{Z}) = \mathbb{P}(\mathbf{Z}|\mathbf{Y})\mathbb{P}(\mathbf{Y})$$

Convention

Notation:

- $\mathbf{Y} = (Y_1, \ldots, Y_n)$ = observed data (typically Speed)
- $\mathbf{Z}$ unobserved data (typically State, for mixture and Hidden Markov model)
- $\boldsymbol{\theta}$= the unknown parameters of $\mathbf{Y}$ and $\mathbf{Z}$.

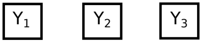
Convention

Notation:

- $\mathbf{Y} = (Y_1, \ldots, Y_n) =$ observed data (typically Speed)
- $\mathbf{Z}$ unobserved data (typically State, for mixture and Hidden Markov model)
- $\boldsymbol{\theta}=$ the unknown parameters of $\mathbf{Y}$ and $\mathbf{Z}$.

Graphical Representation (DAG):

Change point

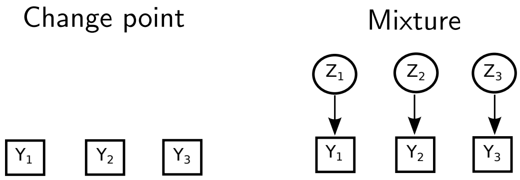


Convention

Notation:

- $\mathbf{Y} = (Y_1, \ldots, Y_n) =$ observed data (typically Speed)
- $\mathbf{Z}$ unobserved data (typically State, for mixture and Hidden Markov model)
- $\boldsymbol{\theta}=$ the unknown parameters of $\mathbf{Y}$ and $\mathbf{Z}$.

Graphical Representation (DAG):

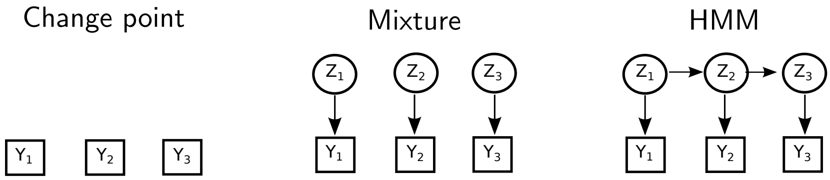


Convention

Notation:

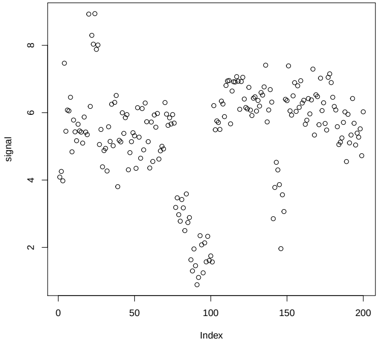
- $\mathbf{Y} = (Y_1, \dots, Y_n)$ = observed data (typically Speed)
- $\mathbf{Z}$ unobserved data (typically State, for mixture and Hidden Markov model)
- $\boldsymbol{\theta}$= the unknown parameters of $\mathbf{Y}$ and $\mathbf{Z}$.

Graphical Representation (DAG):



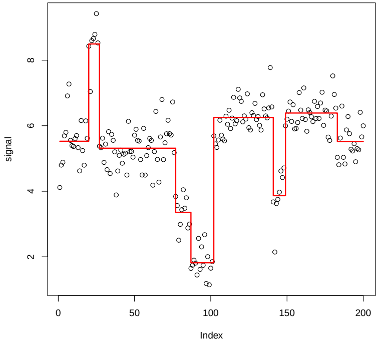
Plan

Change point detection context



Goal : Identifying homogenous region and abrupt changes in the signal.

Change point detection context



These regions may be interpreted afterwards.

Underlying model

Modelling :

- Data $Y_1, \ldots, Y_n$ are drawn from a given pdf, driven by unknown parameter θ

$$Y_i \overset{i.i.d}{\sim} f_\theta(.)$$

θ values change at $K-1$ unknow instants, the change point : $t_1, \ldots, t_{K-1}$:

$$Y_t \sim f(\theta_k) \text{ if } t \text{ in region } I_k = [t_{k-1}+1, t_k]$$

Underlying model

Change point detection in the trend (and/or in variance) :

- Data $Y_1, \ldots, Y_n$ are drawn from a given pdf, driven by unknown parameter θ

$$Y_i \overset{i.i.d}{\sim} f_\theta(.)$$

θ values change at $K-1$ unknow instants, the change point : $t_1, \ldots, t_{K-1}$:

$$Y_t \overset{ind}{\sim} \mathcal{N}(\mu_k, \sigma^2) \text{ if } t \text{ in portion } I_k, \text{ for } k = 1, \ldots, K.$$

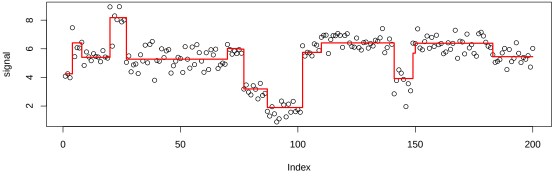
Underlying model

- Data $Y_1, \ldots, Y_n$ are drawn from a given pdf, driven by unknown parameter θ

$$Y_i \overset{i.i.d}{\sim} f_\theta(.)$$

θ values change at $K-1$ unknow instants, the change point : $t_1, \ldots, t_{K-1}$:

$$Y_t \overset{ind}{\sim} \mathcal{N}(\mu_k, \sigma^2) \text{ if } t \text{ in portion } I_k, \text{ for } k = 1, \ldots, K.$$



Remark : $K-1$ change points $\Leftrightarrow$ K regions.

Hybrid algorithm

2 levels of statistical units

- The inference of the breakpoints T is made at the position level t;
- The inference of the groups (status) (Θ, τ_{kp}) is made at the segment level k.

Hybrid algorithm

Alternate parameters estimation with K and P known

1. When $\mathbf{T}$ is fixed, the Expectation-Maximisation (EM) algorithm estimates $\boldsymbol{\theta}$;

$$\hat{\boldsymbol{\theta}}^{(h+1)} = \arg\max_{\boldsymbol{\theta}} \left\{ \log \mathcal{L}_{KP} \left(\boldsymbol{\theta}, T^{(h)} \right) \right\}.$$

$$\log \mathcal{L}_{KP}(\hat{\boldsymbol{\theta}}^{(h+1)}; \hat{\mathbf{T}}^{(h)}) \geq \log \mathcal{L}_{KP}(\hat{\boldsymbol{\theta}}^{(h)}; \hat{\mathbf{T}}^{(h)})$$

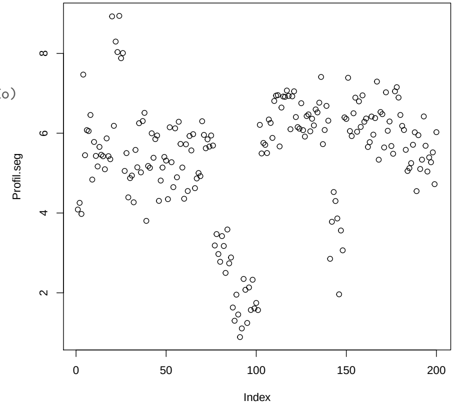
2. When $\boldsymbol{\theta}$ is fixed, dynamic programming estimates $\mathbf{T}$;

$$\hat{\mathbf{T}}^{(h+1)} = \underset{\mathbf{T}}{\mathrm{argmax}} \left\{ \log \mathcal{L}_{KP} \left(\hat{\boldsymbol{\theta}}^{(h+1)}, \mathbf{T} \right) \right\}.$$

$$\log \mathcal{L}_{KP}(\hat{\boldsymbol{\theta}}^{(h+1)}; \hat{\mathbf{T}}^{(h+1)}) \geq \log \mathcal{L}_{KP}(\hat{\boldsymbol{\theta}}^{(h+1)}; \hat{\mathbf{T}}^{(h)})$$

How to perform this segmentation/clustering approach ?

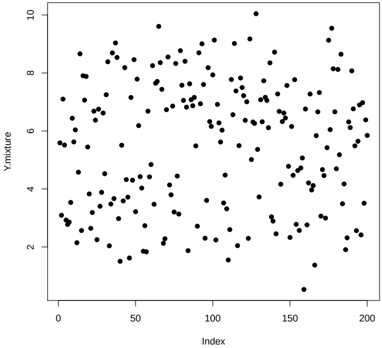
```
## format data into CGHdata
calling(CGHo)<- TRUE ## no classification

Error in calling(CGHo) <- TRUE: object
'CGHo' not found

CGHo@nblevels=2

Error in CGHo@nblevels = 2:  object 'CGHo'
not found

segSignal   <- uniseg(.Object=signalCGH,CGHo=CGHo)

Error in uniseg(.Object = signalCGH, CGHo
= CGHo):  error in evaluating the argument
'.Object' in selecting a method for
function 'uniseg':  Error:  object
'signalCGH' not found

segSignalProf <- getsegprofiles(segSignal)

Error in getsegprofiles(segSignal):  error
in evaluating the argument '.Object' in
selecting a method for function
'getsegprofiles':  Error:  object
'segSignal' not found
```



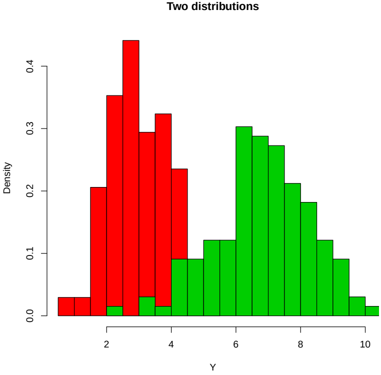
Do it yourself

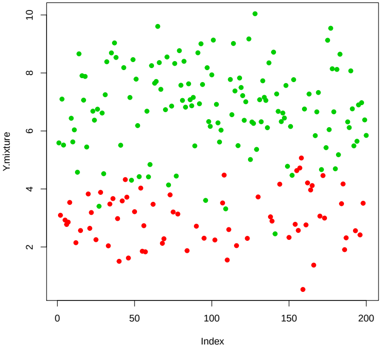
Plan

Problem presentation



Problem presentation





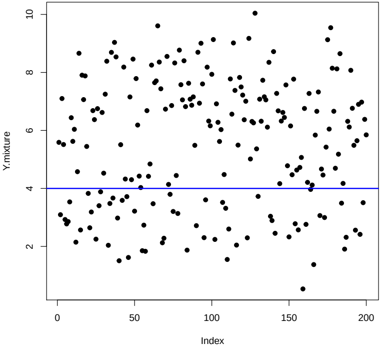
Problem presentation

Basic idea :

"Expert" threshold s

$$State_i = 1 \quad \text{if} Y_i < s$$

$$State_i = 2 \quad \text{if} Y_i \geq s$$



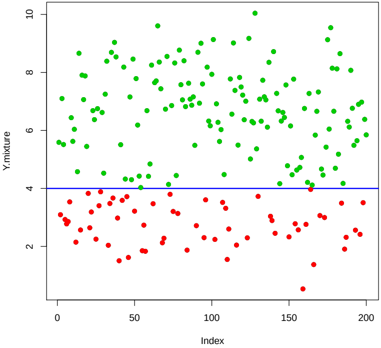
Problem presentation

Basic idea :

"Expert" threshold s

$$State_i = 1 \quad \text{if} Y_i < s$$

$$State_i = 2 \quad \text{if} Y_i \geq s$$



Problem presentation

Basic idea :

"Expert" threshold s

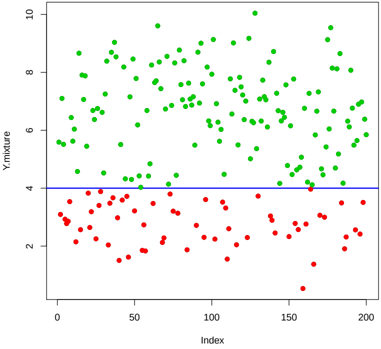
$$State_i = 1 \quad \text{if} Y_i < s$$

$$State_i = 2 \quad \text{if} Y_i \geq s$$

Improvement:

Estimating the threshold s and reconstruction of the hidden state (colour)

Compute the probability to belong to State 1 or 2.



Problem presentation

Basic idea :

"Expert" threshold s

$$State_i = 1 \quad \text{if} Y_i < s$$

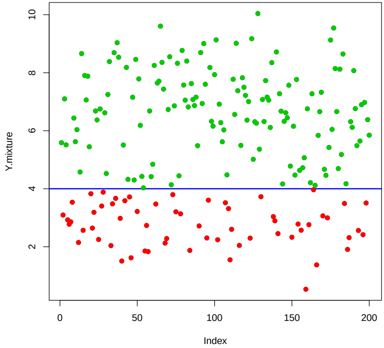
$$State_i = 2 \quad \text{if} Y_i \geq s$$

Improvement:

Estimating the threshold s and reconstruction of the hidden state (colour)

Compute the probability to belong to State 1 or 2.

$\Longrightarrow$ Mixture Model



Proposed model

Model For a given number of states K,

-

Proposed model

Model For a given number of states K,

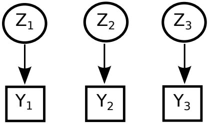
-

$$\text{Model parameters } \boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\gamma})$$

Proposed model

Model For a given number of states K,

- Model parameters $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\gamma})$



Proposed model

Model For a given number of states K,

- Modelling Z: $\pi_k = \mathbb{P}(Z_i = k), \quad k = 1, \ldots, K, \quad \sum_k \pi_k = 1$
  $Z_i \overset{i.i.d}{\sim} \mathcal{M}(1, \boldsymbol{\pi})$
- Modelling Y: The $Y_i's$ are assumed to be independent conditionnaly to $\mathbf{Z}$: $(Y_i|Z_i = k) \overset{i.i.d}{\sim} f_{\gamma_k}()$.

```
K <- 2; N <- 100; mu <- c(3, 7); sigma <- c(1,1.5)
Z <- sample(1:2, size = N, replace=T, prob=c(0.3, 0.7))
plot(Z, col=Z+1, pch=15, cex=0.8)
Y.mixture <- rnorm(N, mean=mu[Z], sd=sigma[Z])
plot(Y.mixture, col=Z+1, pch=19)
```

Proposed model

Model For a given number of states K,

- Modelling Z: $\pi_k = \mathbb{P}(Z_i = k), \quad k = 1, \ldots, K, \quad \sum_k \pi_k = 1$
  $Z_i \overset{i.i.d}{\sim} \mathcal{M}(1, \boldsymbol{\pi})$
- Modelling Y: The $Y_i's$ are assumed to be independent conditionnaly to $\mathbf{Z}$: $(Y_i|Z_i = k) \overset{i.i.d}{\sim} f_{\gamma_k}()$.

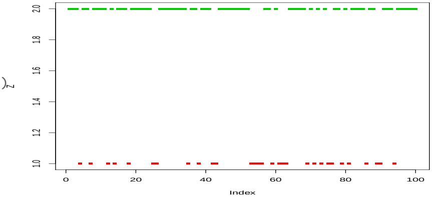
```
K <- 2; N <- 100; mu <- c(3, 7); sigma <- c(1,1.5)
Z <- sample(1:2, size = N, replace=T, prob=c(0.3, 0.7))
plot(Z, col=Z+1, pch=15, cex=0.8)
Y.mixture <- rnorm(N, mean=mu[Z], sd=sigma[Z])
plot(Y.mixture, col=Z+1, pch=19)
```

Proposed model

Model For a given number of states K,

- Modelling Z: $\pi_k = \mathbb{P}(Z_i = k), \quad k = 1, \dots, K, \quad \sum_k \pi_k = 1$
  $Z_i \overset{i.i.d}{\sim} \mathcal{M}(1, \boldsymbol{\pi})$
- Modelling Y: The $Y_i's$ are assumed to be independent conditionnaly to $\mathbf{Z}$: $(Y_i|Z_i = k) \overset{i.i.d}{\sim} f_{\gamma_k}()$.

```
K <- 2; N <- 100; mu <- c(3, 7); sigma <- c(1,1.5)
Z <- sample(1:2, size = N, replace=T, prob=c(0.3, 0.7))
plot(Z, col=Z+1, pch=15, cex=0.8)
Y.mixture <- rnorm(N, mean=mu[Z], sd=sigma[Z])
plot(Y.mixture, col=Z+1, pch=19)
```

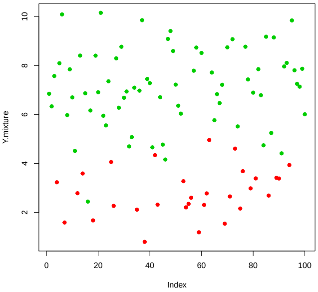


Proposed model

Model For a given number of states K,

- Modelling Z: $\pi_k = \mathbb{P}(Z_i = k), \quad k = 1, \ldots, K, \quad \sum_k \pi_k = 1$
  $Z_i \overset{i.i.d}{\sim} \mathcal{M}(1, \boldsymbol{\pi})$
- Modelling Y: The $Y_i's$ are assumed to be independent conditionnaly to $\mathbf{Z}$: $(Y_i | Z_i = k) \overset{i.i.d}{\sim} f_{\gamma_k}()$.

```
K <- 2; N <- 100; mu <- c(3, 7); sigma <- c(1,1.5)
Z <- sample(1:2, size = N, replace=T, prob=c(0.3, 0.7))
plot(Z, col=Z+1, pch=15, cex=0.8)
Y.mixture <- rnorm(N, mean=mu[Z], sd=sigma[Z])
plot(Y.mixture, col=Z+1, pch=19)
```



Model Properties

- Couples $\{(Y_i, Z_i)\}$ are i.i.d.
- Label switching:
  the model is invariant for any permutation of the labels $\{1, \dots, K\} \Rightarrow$ the mixture model has $K!$ equivalent definitions.
- Distribution of a Y_i:

$$P(Y_i) = \sum_{k=1}^{K} P(Y_i, Z_i = k) = P(Z_i = k) P(Y_i | Z_i = k)$$

- Distribution of $\mathbf{Y}$:

$$\begin{aligned} P(\mathbf{Y}; \boldsymbol{\theta}, \boldsymbol{\pi}) = \prod_{i=1}^{n} \sum_{k=1}^{K} P(Y_i, Z_i = k) &= \prod_{i=1}^{n} \sum_{k=1}^{K} P(Z_i = k) P(Y_i | Z_i = k) \\ &= \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k f_{\gamma_k}(Y_i) \end{aligned}$$

Statistical inference of incomplete data models

Maximum likelihood estimate: We are looking for

$$(\widehat{\boldsymbol{\theta}}, \widehat{\boldsymbol{\pi}}) = \arg\max_{\boldsymbol{\theta}, \boldsymbol{\pi}} \log P(\mathbf{Y}; \boldsymbol{\theta}, \boldsymbol{\pi})$$

- Likelihood of the observed data (or observed likelihood):

$$\log P(\mathbf{Y}; \boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{i=1}^{n} \log \left[\sum_{k=1}^{K} \pi_k f_{\gamma_k}(Y_i) \right]$$

- No analytical estimators.
- Brute force algorithm is not the way

And what if **Z** were observed ?

The complete likelihood is

$$\begin{aligned}\log P(\mathbf{Y}, \mathbf{Z}; \boldsymbol{\theta}, \boldsymbol{\pi}) &= \log P(\mathbf{Z}; \boldsymbol{\pi}) + \log P(\mathbf{Y}|\mathbf{Z}; \boldsymbol{\theta}) \\ &= \sum_i \sum_k Z_{ik} \log \pi_k + \sum_i \sum_k Z_{ik} \log f_{\gamma_k}(Y_i) \\ &= \sum_i \sum_k Z_{ik} [\log \pi_k + \log f_{\gamma_k}(Y_i)].\end{aligned}$$

Now, the sum contains nK (200 if $n = 100$ and $K = 2$) terms. It is much easier.

And what if **Z** were observed ?

The complete likelihood is

$$
\begin{aligned}
\log P(\mathbf{Y}, \mathbf{Z}; \boldsymbol{\theta}, \boldsymbol{\pi}) &= \log P(\mathbf{Z}; \boldsymbol{\pi}) + \log P(\mathbf{Y}|\mathbf{Z}; \boldsymbol{\theta}) \\
&= \sum_i \sum_k Z_{ik} \log \pi_k + \sum_i \sum_k Z_{ik} \log f_{\gamma_k}(Y_i) \\
&= \sum_i \sum_k Z_{ik} [\log \pi_k + \log f_{\gamma_k}(Y_i)].
\end{aligned}
$$

Now, the sum contains nK (200 if $n = 100$ and $K = 2$) terms. It is much easier.

Unfortunately **Z** are unknown.

And what if **Z** were observed ?

The complete likelihood is

$$\begin{aligned}\log P(\mathbf{Y}, \mathbf{Z}; \boldsymbol{\theta}, \boldsymbol{\pi}) &= \log P(\mathbf{Z}; \boldsymbol{\pi}) + \log P(\mathbf{Y}|\mathbf{Z}; \boldsymbol{\theta}) \\ &= \sum_i \sum_k Z_{ik} \log \pi_k + \sum_i \sum_k Z_{ik} \log f_{\gamma_k}(Y_i) \\ &= \sum_i \sum_k Z_{ik} [\log \pi_k + \log f_{\gamma_k}(Y_i)].\end{aligned}$$

Now, the sum contains nK (200 if $n = 100$ and $K = 2$) terms. It is much easier.

Unfortunately **Z** are unknown.

Idea: Replace Z_i, by our best guess, that is :

$$\tau_{ik} := \mathbb{E}(Z_i = k|Y_i) = P(Z_i = k|Y_i)$$

Idea of EM algorithm

Quantity to be maximized :

$$\sum_i \sum_k Z_{ik} [\log \pi_k + \log f_{\gamma_k}(Y_i)].$$

Idea of EM algorithm

Quantity to be maximized :

$$\sum_i \sum_k \tau_{ik} [\log \pi_k + \log f_{\gamma_k}(Y_i)].$$

where τ_{ik} is a good approximation of Z_{ik}.

Idea of EM algorithm

Quantity to be maximized :

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_0) = \sum_i \sum_k \tau_{ik}^{(0)} [\log \pi_k + \log f_{\gamma_k}(Y_i)]$$

where τ_{ik} is a good approximation of Z_{ik}.

If $\boldsymbol{\theta}_0$ is known, a good approximation is
$\tau_{ik}^{(0)} = \mathbb{E}_{\boldsymbol{\theta}_0} \{Z_{ik}|\mathbf{Y}\} = \mathbb{P}_{\boldsymbol{\theta}_0} \{Z_i = k|\mathbf{Y}\}$,

Idea of EM algorithm

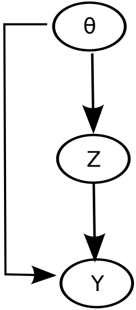
Quantity to be maximized :

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}_0) = \sum_i \sum_k \tau_{ik}^{(0)} [\log \pi_k + \log f_{\gamma_k}(Y_i)]$$

where τ_{ik} is a good approximation of Z_{ik}.

If $\boldsymbol{\theta}_0$ is known, a good approximation is
$\tau_{ik}^{(0)} = \mathbb{E}_{\boldsymbol{\theta}_0}\{Z_{ik}|\mathbf{Y}\} = \mathbb{P}_{\boldsymbol{\theta}_0}\{Z_i = k|\mathbf{Y}\}$,
Idea : iterative algorithm, for a value of $\boldsymbol{\theta}^{(l)}$, compute $\tau_{ik}^{(l)}$, and then update $\boldsymbol{\theta}^{(l)}$ to $\boldsymbol{\theta}^{(l+1)}$.

More generally - EM algorithm



Bayes Formula

$$\begin{aligned} P(\mathbf{Y},\mathbf{Z};\boldsymbol{\theta}) &= P(\mathbf{Y}|\mathbf{Z};\boldsymbol{\theta})P(\mathbf{Z};\boldsymbol{\theta}), \\ &= P(\mathbf{Z}|\mathbf{Y};\boldsymbol{\theta})P(\mathbf{Y};\boldsymbol{\theta}). \end{aligned}$$

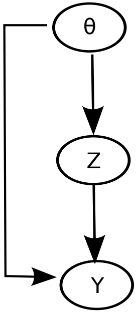
Therefore,

$$\begin{aligned} \log P(\mathbf{Y};\boldsymbol{\theta}) &= \log\left\{P(\mathbf{Y},\mathbf{Z};\boldsymbol{\theta})/P(\mathbf{Z}|\mathbf{Y};\boldsymbol{\theta})\right\} \\ &= \log P(\mathbf{Y},\mathbf{Z};\boldsymbol{\theta}) - \log P(\mathbf{Z}|\mathbf{Y};\boldsymbol{\theta}) \end{aligned}$$

For a given $\boldsymbol{\theta}_0$, we may compute $P_{\boldsymbol{\theta}_0} = P(\mathbf{Z}|\boldsymbol{\theta}_0,\mathbf{Y})$ and

$$\begin{aligned} \log P(\mathbf{Y};\boldsymbol{\theta}) &= \mathbb{E}_{\boldsymbol{\theta}_0}(\log P(\mathbf{Y},\mathbf{Z};\boldsymbol{\theta})) - \mathbb{E}_{\boldsymbol{\theta}_0}(\log P(\mathbf{Z}|\mathbf{Y};\boldsymbol{\theta})) \\ &= Q(\boldsymbol{\theta},\boldsymbol{\theta}_0) - H(\boldsymbol{\theta},\boldsymbol{\theta}_0) \end{aligned}$$

More generally - EM algorithm



Since

$$\log P(\mathbf{Y};\boldsymbol{\theta}) = Q(\boldsymbol{\theta},\boldsymbol{\theta}_0) - H(\boldsymbol{\theta},\boldsymbol{\theta}_0),$$

and $H(\boldsymbol{\theta},\boldsymbol{\theta}_0)$ achieves its maximum in $\boldsymbol{\theta}_0$,

$$\log P(\mathbf{Y};\boldsymbol{\theta}) - \log P(\mathbf{Y};\boldsymbol{\theta}_0) = (Q(\boldsymbol{\theta},\boldsymbol{\theta}_0) - Q(\boldsymbol{\theta},\boldsymbol{\theta}_0)) + (H(\boldsymbol{\theta}_0,\boldsymbol{\theta}_0) - H(\boldsymbol{\theta},\boldsymbol{\theta}_0)).$$

Expectation - Maximization algorithm

1. Phase E :
   Calculate $Q(\boldsymbol{\theta},\boldsymbol{\theta}^k)$ for every $\boldsymbol{\theta}$.
2. Phase M :
   Define $\boldsymbol{\theta}^{k+1} = argmax\, Q(\boldsymbol{\theta},\boldsymbol{\theta}^k)$

EM algorithm for independent mixture model

Recall that $\tau_{ik}^{(\ell)} := P_{\boldsymbol{\theta}^{(\ell)}}(Z_i = k | Y_i)$

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\ell)}) = \sum_i \sum_k \tau_{ik}^{(\ell)} \log \pi_k + \sum_i \sum_k \tau_{ik}^{(\ell)} \log f_{\gamma_k^{(\ell)}}(Y_i)$$

$\rightarrow$ Need to estimate $\tau_{ik}^{(\ell)}$

EM algorithm for independent mixture model

- Initialisation of $\boldsymbol{\theta}^{(0)} = (\pi_1, ..., \pi_K, \gamma_1, ..., \gamma_K)^{(0)}$.

EM algorithm for independent mixture model

- Initialisation of $\boldsymbol{\theta}^{(0)} = (\pi_1, ..., \pi_K, \gamma_1, ..., \gamma_K)^{(0)}$.
- Alternate

  E-step Calculation of

$$\tau_{ik}^{(\ell)} = P(Z_i = k | y_i, \boldsymbol{\theta}^{(\ell-1)}) = \frac{\mathbb{P}(Z_i = k, y_i; \boldsymbol{\theta}^{(\ell-1)})}{\mathbb{P}(y_i, \boldsymbol{\theta}^{(\ell-1)})}$$

EM algorithm for independent mixture model

- Initialisation of $\boldsymbol{\theta}^{(0)} = (\pi_1, ..., \pi_K, \gamma_1, ..., \gamma_K)^{(0)}$.
- Alternate

E-step Calculation of

$$\begin{aligned}
\tau_{ik}^{(\ell)} &= P(Z_i = k | y_i, \boldsymbol{\theta}^{(\ell-1)}) = \frac{\mathbb{P}(Z_i = k, y_i; \boldsymbol{\theta}^{(\ell-1)})}{\mathbb{P}(y_i, \boldsymbol{\theta}^{(\ell-1)})} \\
&= \frac{\mathbb{P}(y_i | Z_i = k; \boldsymbol{\theta}^{(\ell-1)})\mathbb{P}(Z_i = k; \boldsymbol{\theta}^{(\ell-1)})}{\mathbb{P}(y_i; \boldsymbol{\theta}^{(\ell-1)})} \\
&= \frac{\pi_k^{(\ell-1)} f_{\theta_k^{(\ell-1)}}(y_i)}{\sum_{k'} \pi_{k'}^{(\ell-1)} f_{\theta_{k'}^{(\ell-1)}}(y_i)}
\end{aligned}$$

EM algorithm for independent mixture model

- Initialisation of $\boldsymbol{\theta}^{(0)} = (\pi_1, ..., \pi_K, \gamma_1, ..., \gamma_K)^{(0)}$.
- Alternate

E-step Calculation of

$$\begin{aligned}\tau_{ik}^{(\ell)} &= P(Z_i = k | y_i, \boldsymbol{\theta}^{(\ell-1)}) = \frac{\mathbb{P}(Z_i = k, y_i; \boldsymbol{\theta}^{(\ell-1)})}{\mathbb{P}(y_i, \boldsymbol{\theta}^{(\ell-1)})} \\ &= \frac{\mathbb{P}(y_i | Z_i = k; \boldsymbol{\theta}^{(\ell-1)})\mathbb{P}(Z_i = k; \boldsymbol{\theta}^{(\ell-1)})}{\mathbb{P}(y_i; \boldsymbol{\theta}^{(\ell-1)})} \\ &= \frac{\pi_k^{(\ell-1)} f_{\theta_k^{(\ell-1)}}(y_i)}{\sum_{k'} \pi_{k'}^{(\ell-1)} f_{\theta_{k'}^{(\ell-1)}}(y_i)}\end{aligned}$$

M-step Maximization of

$$(\boldsymbol{\pi}, \gamma) \longmapsto \sum_i \sum_k \tau_{ik}^{(\ell)} [\log \pi_k + \log f(x_i; \gamma_k)]$$

In the example

- $Z \in \{1,2\}$: $P(Z=1)=\pi_1$ and $P(Z=2)=1-\pi_1$
- For $k=1$ or 2, $(X|Z=k) \sim \mathcal{N}(\mu_k, \sigma_k^2)$
- The parameter vector is $\boldsymbol{\theta} = (\pi, \mu_1, \mu_2, \sigma_2^2, \sigma_2^2)$

$$
\begin{aligned}
\widehat{\pi}_1^{(\ell+1)} &= \frac{1}{n}\sum_{i=1}^{n} \tau_{i1}^{(\ell)}, \\
\widehat{\mu}_k^{(\ell+1)} &= \frac{1}{\sum_{i=1}^{n} \tau_{ik}^{(\ell)}} \sum_{i=1}^{n} \tau_{ik}^{(\ell)} y_i \\
\widehat{\sigma}^2_{k^{(\ell+1)}} &= \frac{1}{\sum_{i=1}^{n} \tau_{ik}^{(\ell)}} \sum_{i=1}^{n} \tau_{ik}^{(\ell)} (y_i - \widehat{\mu}_k^{(\ell)})^2
\end{aligned}
$$

$\rightarrow$ They are a weighted version of the usual maximum likelihood estimates.

Back on earth - Practically speaking

```
#library('mclust')
library('mixtools')
Y.clustering <- normalmixEM (Y.mixture, lambda = NULL, mu = NULL, sigma = NULL, k = 2,
                             mean.constr = NULL, sd.constr = NULL,
                             epsilon = 1e-07, maxit = 1000, maxrestarts=20)
```

```
number of iterations= 94
```

```
summary(Y.clustering)
```

```
summary of normalmixEM object:
         comp 1
lambda 0.644837
mu     7.366618
sigma  1.374730
         comp 2
lambda 0.355163
mu     3.072069
sigma  1.142811
loglik at estimate:  -220.6712
```

```
Y.clustering$posterior
```

```
             comp.1
  [1,] 0.9969687292
  [2,] 0.9851514346
```

Do it yourself

Reconstruction of hidden state **Z**

Since $\pi_{ik} = \mathbb{P}(Z_i = k)$, and $\hat{\pi}_{ik}$ is an estimation of π_{ik},

$$\widehat{Z}_i = \operatorname{argmax} k = 1, K = \widehat{\pi}_{ik}$$

Plan

Markov chain model

Modelling the dependence in state sequence: If an animal is feeding at time i, he has more chance to be feeding at time $i+1$ than if he was travelling at time i.

$$P(Z_{i+1} = 1 | Z_i = 1) \neq P(Z_{i+1} = 1 | Z_i = 2)$$

Markov Chain definition $\mathbf{Z}$ is a Markov chain if

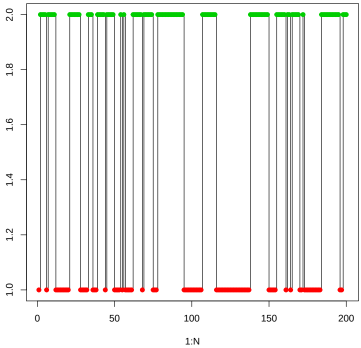
$$P(Z_{i+1} | Z_{1:i}) = P(Z_{i+1} | Z_i)$$

$\mathbf{Z}$ is completely defined by the distribution $\nu_1 = P(Z_1)$ and the transition matrix

$$\Pi = \begin{bmatrix} \pi_{11} & 1 - \pi_{11} \\ 1 - \pi_{22} & 1 - \pi_{22} \end{bmatrix}$$

Markov chain simulation

```r
### Hidden State simulation
set.seed(6)
N <- 200
pi11 <- 0.8
pi22 <- 0.9
## initial distribution
mu1 <- c(0.5, 0.5)
##transition matrix
PI <- matrix(c(pi11, 1-pi11, 1-pi22, pi22), ncol=2, byrow = T)

##initialisation of Z
Z <- rep(NA, N)
Z[1] <- sample(1:2, size=1, prob = mu1)
for( i in 1:(N-1))
{
  Z[i+1] <- sample(1:2, size=1, prob = PI[Z[i],])
}
plot(1:N, Z, "s")
points(1:N, Z, col=Z+1, pch=19)
```



Hidden Markov Chain model

Model For a given number of states K,

- Hidden States $\mathbf{Z}$ model: $\mathbf{Z}$ is assumed to follow a Markov Chain model with unknown initial distribution $\boldsymbol{\nu}$ and transition matrix $\boldsymbol{\Pi}$.
- Observations $\mathbf{Y}$ model: The $Y_i's$ are assumed to be independent conditionnaly to $\mathbf{Z}$: $(Y_i|Z_i = k) \overset{i.i.d}{\sim} f_{\gamma_k}()$.

Hidden Markov Chain model

Model For a given number of states K,

- Hidden States $\mathbf{Z}$ model: $\mathbf{Z}$ is assumed to follow a Markov Chain model with unknown initial distribution $\boldsymbol{\nu}$ and transition matrix $\mathbf{\Pi}$.
- Observations $\mathbf{Y}$ model: The $Y_i's$ are assumed to be independent conditionnaly to $\mathbf{Z}$: $(Y_i|Z_i = k) \overset{i.i.d}{\sim} f_{\gamma_k}()$.

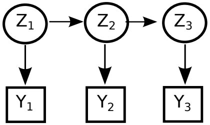
Model parameters are $\boldsymbol{\theta} = (\boldsymbol{\nu}, \mathbf{\Pi}, \gamma)$

Hidden Markov Chain model

Model For a given number of states K,

- Hidden States $\mathbf{Z}$ model: $\mathbf{Z}$ is assumed to follow a Markov Chain model with unknown initial distribution $\boldsymbol{\nu}$ and transition matrix $\mathbf{\Pi}$.
- Observations $\mathbf{Y}$ model: The $Y_i's$ are assumed to be independent conditionnaly to $\mathbf{Z}$: $(Y_i|Z_i = k) \overset{i.i.d}{\sim} f_{\gamma_k}()$.

Model parameters are $\boldsymbol{\theta} = (\boldsymbol{\nu}, \mathbf{\Pi}, \gamma)$

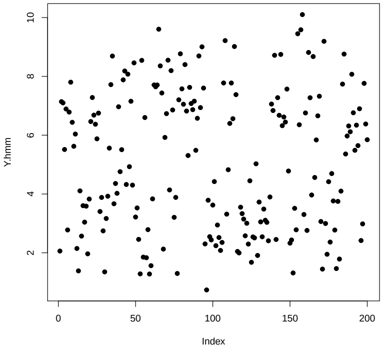


Hidden Markov Chain simulation

```
### observation simulation
mu <- c(3, 7)
sigma <- c(1,1.5)
Y.hmm <- rnorm(N, mean=mu[Z], sd=sigma[Z])
plot(Y.hmm, pch=19)
plot(Y.hmm, pch=19, col=Z+1)
```

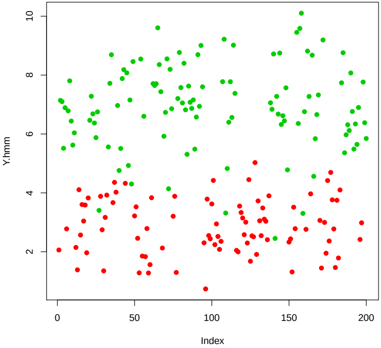
Hidden Markov Chain simulation

```r
### observation simulation
mu <- c(3, 7)
sigma <- c(1,1.5)
Y.hmm <- rnorm(N, mean=mu[Z], sd=sigma[Z])
plot(Y.hmm, pch=19)
plot(Y.hmm, pch=19, col=Z+1)
```



Hidden Markov Chain simulation

```
### observation simulation
mu <- c(3, 7)
sigma <- c(1,1.5)
Y.hmm <- rnorm(N, mean=mu[Z], sd=sigma[Z])
plot(Y.hmm, pch=19)
plot(Y.hmm, pch=19, col=Z+1)
```



Sojourn time properties

T_i, the sojourn time in State i follows a geometric distribution

$$\mathbb{P}(T_i = l) = (\Pi_{ii})^{l-1}(1 - \Pi_{ii})$$

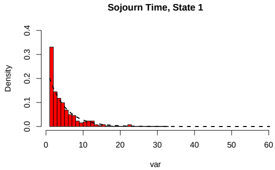
```r
switchTime <- which(diff(Z)!=0)
sojournTime <- diff(switchTime)
sojournState <- rep(c(3-Z[1], Z[1]),
                    length.out = length(sojournTime))
br <- unique(quantile(sojournTime,
                      p<- seq(1/N, 1, length.out = 6)))

abc <- seq(1, max(sojournTime)+10)
lapply(1:2, function(i){
  var <- sojournTime[sojournState==i]
  br <- sort(unique(var))
  hist(var, col=i+1, freq=F,
       xlim=range(sojournTime),
       ylim=c(0,0.4),
       main=paste0("Sojourn Time, State ", i),
       breaks=br)
  lines(abc, dgeom(abc-1, prob = 1-PI[i,i]),
        col=1, lwd=2, lty = 1+i)
})
```

Sojourn time properties

T_i, the sojourn time in State i follows a geometric distribution

$$\mathbb{P}(T_i = l) = (\Pi_{ii})^{l-1}(1 - \Pi_{ii})$$

```
switchTime <- which(diff(Z)!=0)
sojournTime <- diff(switchTime)
sojournState <- rep(c(3-Z[1], Z[1]),
                    length.out = length(sojournTime))
br <- unique(quantile(sojournTime,
                      p<- seq(1/N, 1, length.out = 6)))

abc <- seq(1, max(sojournTime)+10)
lapply(1:2, function(i){
  var <- sojournTime[sojournState==i]
  br <- sort(unique(var))
  hist(var, col=i+1, freq=F,
       xlim=range(sojournTime),
       ylim=c(0,0.4),
       main=paste0("Sojourn Time, State ", i),
       breaks=br)
  lines(abc, dgeom(abc-1, prob = 1-PI[i,i]),
        col=1, lwd=2, lty = 1+i)
})
```

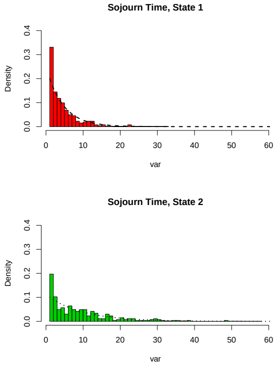


Sojourn time properties

T_i, the sojourn time in State i follows a geometric distribution

$$\mathbb{P}(T_i = l) = (\Pi_{ii})^{l-1}(1 - \Pi_{ii})$$

```
switchTime <- which(diff(Z)!=0)
sojournTime <- diff(switchTime)
sojournState <- rep(c(3-Z[1], Z[1]),
                    length.out = length(sojournTime))
br <- unique(quantile(sojournTime,
                      p<- seq(1/N, 1, length.out = 6)))

abc <- seq(1, max(sojournTime)+10)
lapply(1:2, function(i){
  var <- sojournTime[sojournState==i]
  br <- sort(unique(var))
  hist(var, col=i+1, freq=F,
       xlim=range(sojournTime),
       ylim=c(0,0.4),
       main=paste0("Sojourn Time, State ", i),
       breaks=br)
  lines(abc, dgeom(abc-1, prob = 1-PI[i,i]),
        col=1, lwd=2, lty = 1+i)
})
```

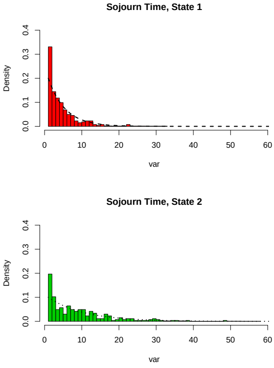


Sojourn time properties

T_i, the sojourn time in State i follows a geometric distribution

$$\mathbb{P}(T_i = l) = (\Pi_{ii})^{l-1}(1 - \Pi_{ii})$$

```r
switchTime <- which(diff(Z)!=0)
sojournTime <- diff(switchTime)
sojournState <- rep(c(3-Z[1], Z[1]),
                    length.out = length(sojournTime))
br <- unique(quantile(sojournTime,
                      p<- seq(1/N, 1, length.out = 6)))

abc <- seq(1, max(sojournTime)+10)
lapply(1:2, function(i){
  var <- sojournTime[sojournState==i]
  br <- sort(unique(var))
  hist(var, col=i+1, freq=F,
       xlim=range(sojournTime),
       ylim=c(0,0.4),
       main=paste0("Sojourn Time, State ", i),
       breaks=br)
  lines(abc, dgeom(abc-1, prob = 1-PI[i,i]),
        col=1, lwd=2, lty = 1+i)
})
```



see Semi Hidden Markov Model for removing this assumption

Statistical inference of incomplete data models

Maximum likelihood estimate: We are looking for

$$(\widehat{\gamma}, \widehat{\boldsymbol{\Pi}}, \widehat{\boldsymbol{\nu}}) = \arg\max_{\boldsymbol{\theta}, \boldsymbol{\Pi}, \boldsymbol{\nu}} \log P(\mathbf{Y}; \boldsymbol{\theta}, \boldsymbol{\Pi}, \boldsymbol{\nu})$$

$$\begin{aligned}
\log P(\mathbf{X}, \mathbf{Z}; \boldsymbol{\theta}) \quad = \quad & \sum_k Z_{1k} \log \nu_k \\
& + \sum_{i>1} \sum_{k,\ell} Z_{i-1,k} Z_{i,\ell} \log \pi_{k\ell} \\
& + \sum_i \sum_k Z_{ik} \log f(X_i; \gamma_k)
\end{aligned}$$

Statistical inference of incomplete data models

Maximum likelihood estimate: We are looking for

$$(\widehat{\gamma}, \widehat{\boldsymbol{\Pi}}, \widehat{\boldsymbol{\nu}}) = \arg \max_{\boldsymbol{\theta}, \boldsymbol{\Pi}, \boldsymbol{\nu}} \log P(\mathbf{Y}; \boldsymbol{\theta}, \boldsymbol{\Pi}, \boldsymbol{\nu})$$

$$\begin{aligned}
\mathbb{E}\left(\log P(\mathbf{X}, \mathbf{Z}) | Y_{1:N}\right) \quad = \quad & \sum_k \mathbb{E}\left(Z_{1k} | Y_{1:N}\right) \log \nu_k \\
& + \sum_{i>1} \sum_{k,\ell} \mathbb{E}\left(Z_{i-1,k} Z_{i,\ell} | Y_{1:N}\right) \log \pi_{k\ell} \\
& + \sum_i \sum_k \mathbb{E}\left(Z_{ik} | Y_{1:N}\right) \log f\left(X_i; \gamma_k\right)
\end{aligned}$$

Statistical inference of incomplete data models

Maximum likelihood estimate: We are looking for

$$(\widehat{\gamma}, \widehat{\mathbf{\Pi}}, \widehat{\boldsymbol{\nu}}) = \arg\max_{\boldsymbol{\theta}, \mathbf{\Pi}, \boldsymbol{\nu}} \log P(\mathbf{Y}; \boldsymbol{\theta}, \mathbf{\Pi}, \boldsymbol{\nu})$$

$$\begin{aligned}
\mathbb{E}\left(\log P(\mathbf{X}, \mathbf{Z}) | Y_{1:N}\right) \quad = \quad & \sum_k \mathbb{P}\left(Z_1 = k | Y_{1:N}\right) \log \nu_k \\
& + \sum_{i>1} \sum_{k,\ell} \mathbb{P}\left(Z_{i-1} = k, Z_i = l | Y_{1:N}\right) \log \pi_{k\ell} \\
& + \sum_i \sum_k \mathbb{P}\left(Z_i = k | Y_{1:N}\right) \log f\left(X_i; \gamma_k\right)
\end{aligned}$$

EM Algorithm (Baum Welch)

- Initialisation of $\boldsymbol{\theta}^{(0)} = (\Pi, \gamma_1, ..., \gamma_K)^{(0)}$.
- While the convergence is not reached

E-step Calculation of

$$\begin{aligned}\tau_{ik}^{(\ell)} &= P(Z_i = k | Y_{1:n}, \boldsymbol{\theta}^{(\ell-1)}) \\ \eta_{ikh}^{(\ell)} &= \mathbb{E}[Z_{i-1,k} Z_{ih} | Y_{1:n}, \boldsymbol{\theta}^{(\ell-1)}]\end{aligned}$$

M-step Maximization in $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\gamma})$ of

$$\sum_k \tau_{1k}^{(\ell)} \log \nu_k + \sum_{i>1} \sum_{k,h} \eta_{ikh}^{(\ell)} \log \pi_{kh} + \sum_i \sum_k \tau_{ik}^{(\ell)} \log f(x_i; \gamma_k)$$

EM Algorithm (Baum Welch)

- Initialisation of $\boldsymbol{\theta}^{(0)} = (\Pi, \gamma_1, ..., \gamma_K)^{(0)}$.
- While the convergence is not reached

E-step Calculation of Smart algorithm Forward-Backward algorithm

M-step Maximization in $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\gamma})$ of

$$\sum_k \tau_{1k}^{(\ell)} \log \nu_k + \sum_{i>1} \sum_{k,h} \eta_{ikh}^{(\ell)} \log \pi_{kh} + \sum_i \sum_k \tau_{ik}^{(\ell)} \log f(x_i; \gamma_k)$$

Reconstruction of hidden state $\mathbf{Z}$

Most credible value for Z_i: We are interested in

$$\underset{k}{\operatorname{argmax}}\, \mathbb{P}(Z_i = k | Y_{1:n}) = \underset{k}{\operatorname{argmax}}\, \tau_{ik}.$$

Reconstruction of hidden state **Z**

Most credible value for Z_i: We are interested in

$$\underset{k}{\operatorname{argmax}}\, \mathbb{P}(Z_i = k | Y_{1:n}) = \underset{k}{\operatorname{argmax}}\, \tau_{ik}.$$

Most credible sequence for **Z** We are interested in

$$\underset{k_1,\ldots k_n}{\operatorname{argmax}}\, \mathbb{P}(Z_1 = k_1, \ldots Z_n = k_n | Y_{1:n}) = ???$$

Reconstruction of hidden state **Z**

Most credible value for Z_i: We are interested in

$$\underset{k}{\operatorname{argmax}}\, \mathbb{P}(Z_i = k | Y_{1:n}) = \underset{k}{\operatorname{argmax}}\, \tau_{ik}.$$

Most credible sequence for **Z** We are interested in

$$\underset{k_1,\ldots k_n}{\operatorname{argmax}}\, \mathbb{P}(Z_1 = k_1, \ldots Z_n = k_n | Y_{1:n}) = ???$$

But force brut algorithm is not possible

$\longrightarrow$ a smart algorithm : Viterbi algorithm

Viterbi algorithm

Key quantity: The probability of the best hidden path from time 1 to i who finished in k

Viterbi algorithm

Key quantity:

$$\delta_i(k) = \max_{k_1,\ldots k_{i-1}} \mathbb{P}(Y_{1:i}, Z_1 = k_1, \ldots, Z_{i-1} = k_{i-1}, Z_i = k)$$

Viterbi algorithm

Key quantity:

$$\delta_i(k) = \max_{k_1,\dots k_{i-1}} \mathbb{P}(Y_{1:i}, Z_1 = k_1, \dots, Z_{i-1} = k_{i-1}, Z_i = k)$$

- *Initialisation*

$$\begin{aligned} \delta_1(k) &= \mathbb{P}(Z_1 = k, Y_1) = \mathbb{P}(Z_1 = k)\mathbb{P}(Y_1 | Z_1 = k) = \nu(k) f_{\gamma_k}(y_1) \\ \psi_1(k) &= 0 \end{aligned}$$

Viterbi algorithm

Key quantity:

$$\delta_i(k) = \max_{k_1,\dots k_{i-1}} \mathbb{P}(Y_{1:i}, Z_1 = k_1, \dots, Z_{i-1} = k_{i-1}, Z_i = k)$$

- *Initialisation*

$$\begin{array}{rcl} \delta_1(k) & = & \mathbb{P}(Z_1 = k, Y_1) = \mathbb{P}(Z_1 = k)\mathbb{P}(Y_1 | Z_1 = k) = \nu(k) f_{\gamma_k}(y_1) \\ \psi_1(k) & = & 0 \end{array}$$

- *Recurrence, for $i = 1, \dots n-1$*

$$\begin{array}{rcl} \delta_{i+1}(k) & = & \max_{k_1,\dots,k_i} \mathbb{P}(Y_{1:i}, Z_1 = k_1, \dots, Z_{i-1} = k_{i-1}, Z_i = k_i, Z_{i+1} = k) \\ & = & \max_{k_i} \{\delta_i(k_i)\mathbb{P}(Y_i | Z_i = k_i)\mathbb{P}(Z_{i+1} | Z_i = k_i)\} \\ \psi_{i+1}(k) & = & \operatorname{argmax}_j \{\delta_i j \Pi_{jk}\} \end{array}$$

Estimation of HMM with R

```
library('depmixS4')

df <- data.frame(Y=Y)
K=2
m1 <- depmix(Y~1,data=df, nstates=2, family=gaussian())
fit.model <- fit(m1)
summary(fit.model)
Z.hat <- viterbi(fit.model)[,1]
table(Z, Z.hat)
```

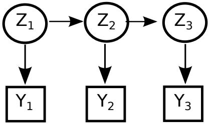
State Space model

This is the trendy name for HMM, with potentially continuous value for $\mathbf{Z}$.

State Space model

This is the trendy name for HMM, with potentially continuous value for **Z**.

- **Z** is a sequence of hidden states and the observations **Y** are ruled by this sequence.

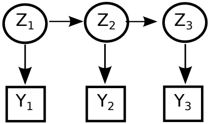


Z could be thougth as the actual locations and **Y** the observed locations.

State Space model

This is the trendy name for HMM, with potentially continuous value for **Z**.

- **Z** is a sequence of hidden states and the observations **Y** are ruled by this sequence.



Z could be thougth as the actual locations and **Y** the observed locations.

- Estimation tools are EM algorithm or Bayesian framework (with monte carlo based estimation technics)

Plan

And more ...

- It is possible to include dependency in $\mathbf{Y}$.
- Markovian property could be removed (SHMM)
- Presented methods may be used on several signals (ex *Speed* and *angle*)
- Work in progress for continuous time Markov model and dependent observations.

But

Limitations

Trajectories are in continuous space and continuous time.

- Mostly, discrete time : effects of the sampling step and assumption of regularity.
- Segmentation methods consider a signal in time,
- Spatial information is lost.
- Those methods are useful to identify different regimes of movement. This difference may be due to behaviour or the environment or interaction.

Many thanks to Emilie Lebarbier, Marie-Laure Martin-Magniette, Stéphane Robin for some contents of the slides.

Many thanks to Andrea and Linda for the invitation and organisation.