Model based detection of homogeneous portions in trajectories

Marie-Pierre Etienne

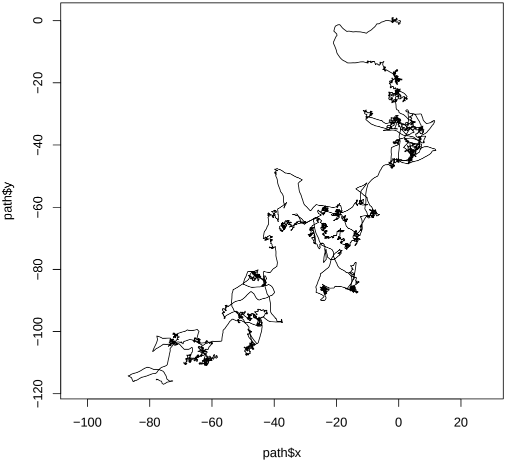
AgroParisTech / INRA

Movement Ecology Workshop 2015 - Port Elizabeth

Detection of homogenous regions in trajectories

Why ?

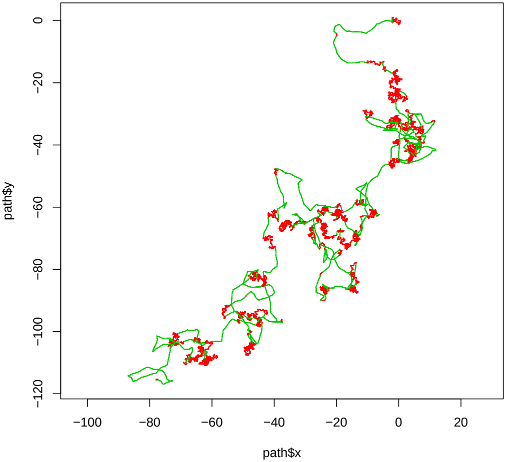
- Different behaviours.
- Link with different activities.
- Link with different environmental conditions.



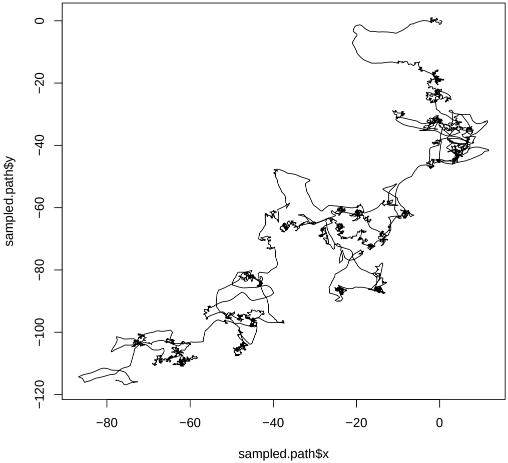
Detection of homogenous regions in trajectories

Why ?

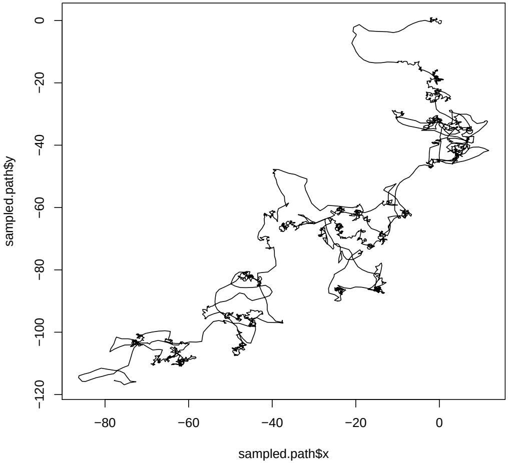
- Different behaviours.
- Link with different activities.
- Link with different environmental conditions.



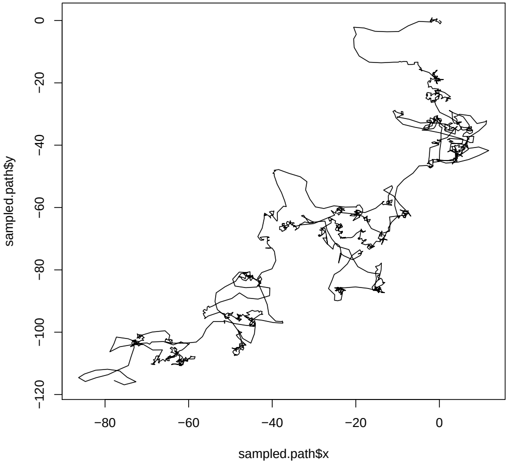
Effect of sampling step



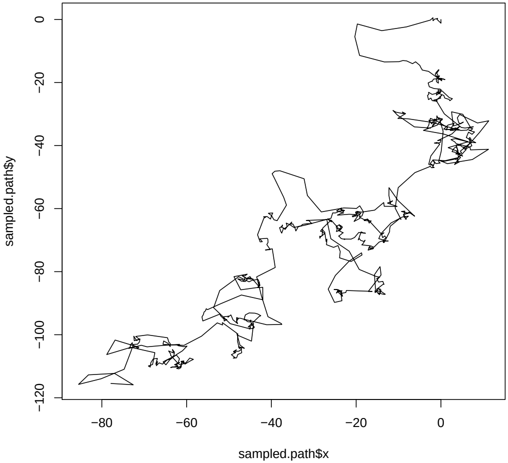
Effect of sampling step



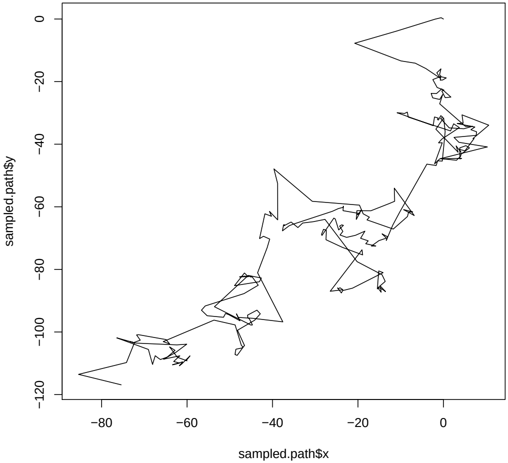
Effect of sampling step



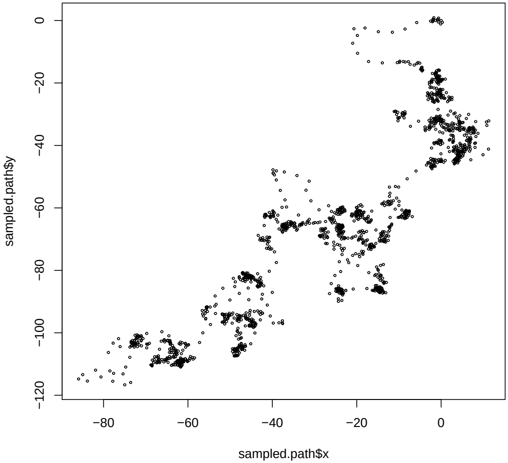
Effect of sampling step



Effect of sampling step



Effect of sampling step



Summarising trajectories

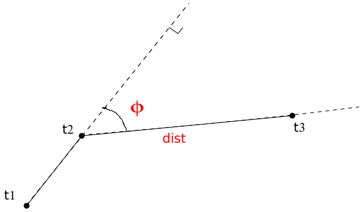
$(t_1, \ldots, t_N)$ denotes the time acquisition and $((x_1, y_1), \ldots, (x_N, y_N))$ the position at those times.

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.

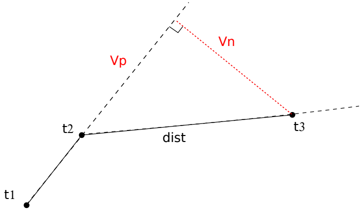
Trajetories as Persistent and Normal Velocity

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

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

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.

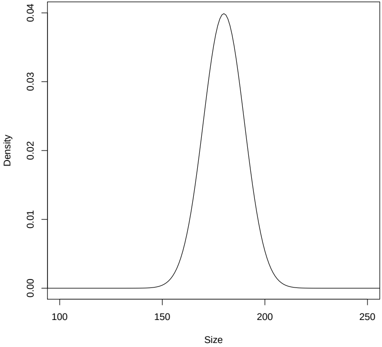
Model and parameters

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)

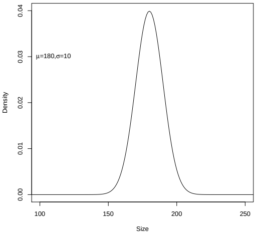
$$Y_i \overset{i.i.d}{\sim} \mathcal{N}(\mu, \sigma)$$

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.

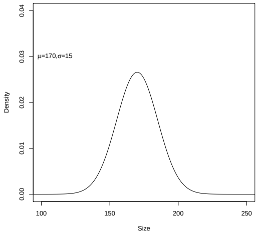


Model and parameters

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, θ) 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 θ ?
- Information on $\mathbf{Z}$: is it possible to "reconstruct" the unobserved data $\mathbf{Z}$?

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.
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

Notations:

- $\mathbf{Y} = (Y_1, \dots, Y_n)$ = observed data (typically Speed)
- $\mathbf{Z} = (Z_1, \dots, Z_n)$ unobserved data (typically State, for mixture and Hidden Markov model)
- $\boldsymbol{\theta}$= the unknown parameters of $\mathbf{Y}$ and $\mathbf{Z}$.

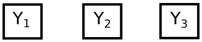
Convention

Notations:

- $\mathbf{Y} = (Y_1, \ldots, Y_n) =$ observed data (typically Speed)
- $\mathbf{Z} = (Z_1, \ldots, Z_n)$ 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

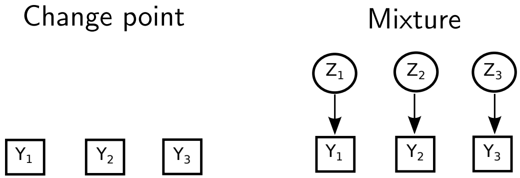


Convention

Notations:

- $\mathbf{Y} = (Y_1, \ldots, Y_n) =$ observed data (typically Speed)
- $\mathbf{Z} = (Z_1, \ldots, Z_n)$ unobserved data (typically State, for mixture and Hidden Markov model)
- $\boldsymbol{\theta}=$ the unknown parameters of $\mathbf{Y}$ and $\mathbf{Z}$.

Graphical Representation (DAG):

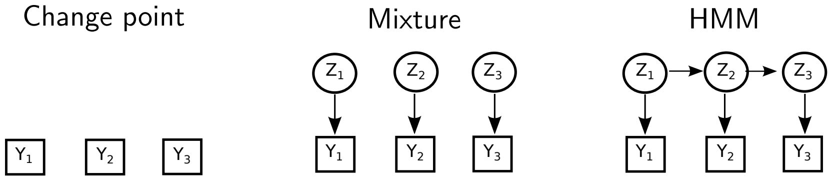


Convention

Notations:

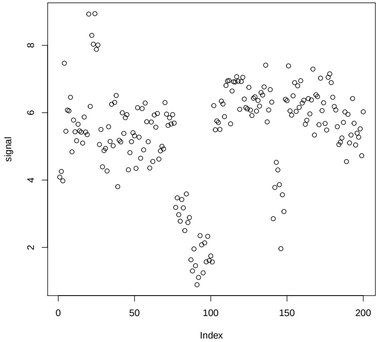
- $\mathbf{Y} = (Y_1, \dots, Y_n) =$ observed data (typically Speed)
- $\mathbf{Z} = (Z_1, \dots, Z_n)$ unobserved data (typically State, for mixture and Hidden Markov model)
- $\boldsymbol{\theta}=$ the unknown parameters of $\mathbf{Y}$ and $\mathbf{Z}$.

Graphical Representation (DAG):



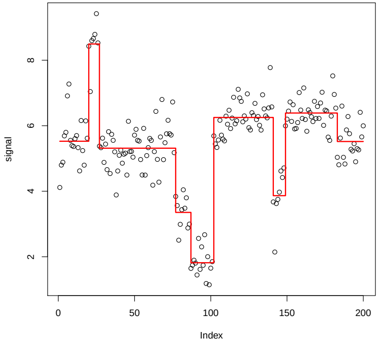
Plan

Change point detection context



Goal: Identifying homogenous regions 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 : $t_1, \ldots, t_{K-1}$:

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

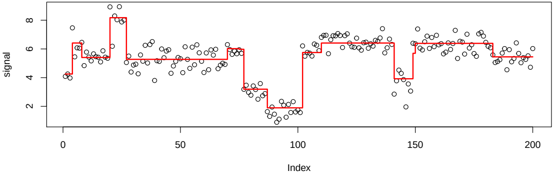
Underlying model

- 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.

Estimation procedure

- Unknown parameters : $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$, σ, and $\mathbf{T} = (T_1, \ldots, T_K)$, but also K itself.

Estimation procedure

- Unknown parameters : $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$, σ, and $\mathbf{T} = (T_1, \ldots, T_K)$, but also K itself.

Estimation procedure

- Unknown parameters : $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$, σ, and $\mathbf{T} = (T_1, \ldots, T_K)$, but also K itself.
- Estimation Procedure
  - For given K and $\mathbf{T}$, $\boldsymbol{\theta}$ is estimated using maximum likelihood.

Estimation procedure

- Unknown parameters : $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$, σ, and $\mathbf{T} = (T_1, \ldots, T_K)$, but also K itself.
- Estimation Procedure
  - For given K, compute the maximum likelihood for any possible position for $\mathbf{T}$, But,

$$\binom{n-1}{K-1}$$

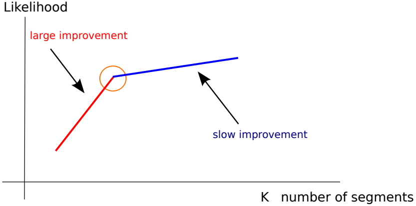
possible choices for the $K-1$ positions, that is 10^{30} for $K = 10$, $n = 200$, ($\approx 10^{11}$ years on a 2014 computer).

$\Rightarrow$ Practically impossible even for small K and n

Dynamic programming

Estimation procedure

- Unknown parameters : $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$, σ, and $\mathbf{T} = (T_1, \ldots, T_K)$, but also K itself.
- Estimation Procedure
  - Estimating K. Likelihood increases with the number of segment K, use a penalized likelihood criterion and



Estimation procedure

Likelihood

$$
\begin{aligned}
2log(P_K(\mathbf{Y}, \mathbf{T}, \boldsymbol{\theta})) &= 2\sum_{k=1}^{K} \log f(\{Y_t\}_{t\in I_k}; \theta_k) = 2\sum_{k=1}^{K}\sum_{t\in I_k} \log f(Y_t; \theta_k) \\
&= -n\log\sigma^2 - \frac{1}{\sigma^2}\sum_{k=1}^{K}\sum_{t\in I_k}(Y_t - \mu_k)^2 + \text{cst.}
\end{aligned}
$$

Estimations

$$
(\hat{\mathbf{T}}, \hat{\boldsymbol{\theta}}) = \underset{(\mathbf{T},\boldsymbol{\theta})}{\operatorname{argmax}}\, log(P_K(\mathbf{Y}, \mathbf{T}, \boldsymbol{\theta}))
$$

If the change points are known

$$
\widehat{\mu}_k = \frac{1}{n_k}\sum_{t\in I_k} Y_t \qquad\qquad \widehat{\sigma}^2 = \frac{1}{n}\sum_{k=1}^{K}\sum_{t\in I_k}(Y_t - \widehat{\mu}_k)^2
$$

Finding the K-1 change points

Considering all possible segmentations, the best segmentation minimizes

$$J_k(1, n) = \sum_{k=1}^{K} \sum_{t \in I_k} (Y_t - \widehat{\mu}_k)^2.$$

Finding the K-1 change points

Considering all possible segmentations, the best segmentation minimizes

$$J_k(1, n) = \sum_{k=1}^{K} \sum_{t \in I_k} (Y_t - \widehat{\mu}_k)^2.$$

Dynamic programming, with complexity ($\mathcal{O}(n^2)$).

Finding the K-1 change points

Considering all possible segmentations, the best segmentation minimizes

$$J_k(1,n) = \sum_{k=1}^{K} \sum_{t \in I_k} (Y_t - \widehat{\mu}_k)^2.$$

Dynamic programming, with complexity ($\mathcal{O}(n^2)$).

Finding the K-1 change points

Sub-paths of the optimal path are themselves optimal, Bellmann optimality

Initialisation: Compute for $0 \leq i < j \leq n$, cost of portion I_{ij} :

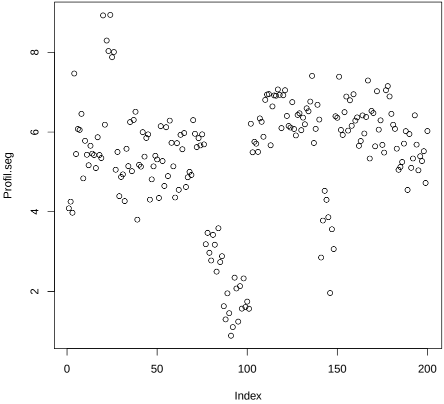
$$J_1(i,j) = \sum_{t=i+1}^{j} (Y_t - \widehat{\mu})^2$$

Etape k: Compute for $2 \leq k \leq K$, $J_k(i,j)$ the cost of the best segmentation in k segments between i and j.

$$J_k(i,j) = \min_{i<h<j} \left[J_{k-1}(i,h) + J_1(h+1,j) \right].$$

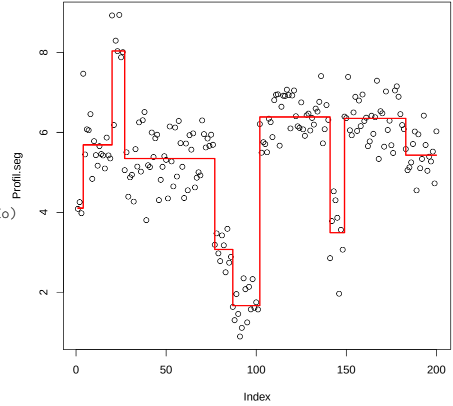
How to perform this segmentation approach ?

```
load("../Data/dataSegmentation.Rd")
summary(Profil.seg)


   Min. 1st Qu.
 0.8904  4.8720
 Median    Mean
 5.6990  5.3660
3rd Qu.    Max.
 6.3070  8.9400


plot(Profil.seg)
```



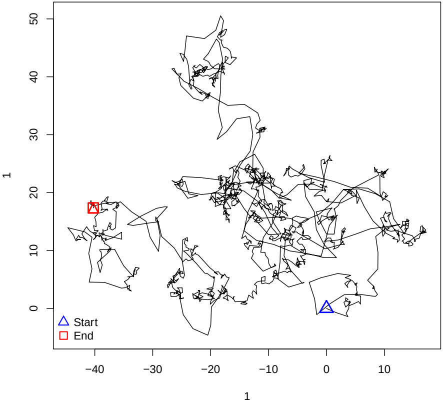
How to perform this segmentation approach ?

```r
library('cghseg')
## format data into CGHdata
signalCGH    <- new("CGHdata",Y=Profil.seg)
CGHo         <- new("CGHoptions")
calling(CGHo)<- FALSE ## no classification

segSignal    <- uniseg(.Object=signalCGH,CGHo=CGHo)
segSignalProf <- getsegprofiles(segSignal)
plot(Profil.seg)
lines(1:length(segSignalProf),
      segSignalProf, type="s", col=2, lwd=2)
```

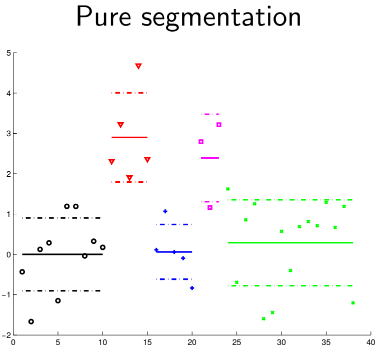


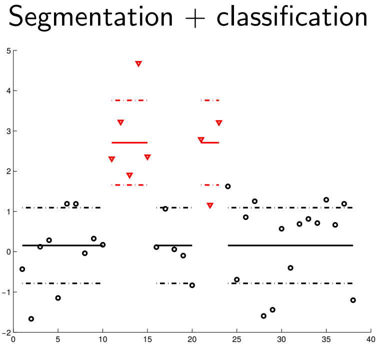
Do it yourself

```
load(file="../Data/trajEx.Rd")
plot(traj.ex, addpoints = F)
legend("bottomleft",pch=c(2, 0), col=c(4,2),
       legend=c("Start", "End"), bty = "n",
       pt.lwd = c(1.5,1.5), pt.cex = c(1.5,1.5))
```



When Segmentation is not sufficient - clustering segmentation model





Segmentation-Clustering

- The distribution of the signal given the group of the segment is

$$
\begin{aligned}
t \in I_k, k \in p \qquad \Rightarrow \qquad & Y_t \sim \mathcal{N}(m_p, \sigma^2) \\
Y^k | Z_{kp} = 1 \sim \mathcal{N}(m_p, \sigma^2).
\end{aligned}
$$

- It is a model of segmentation / clustering.

Segmentation-Clustering

- The distribution of the signal given the group of the segment is

$$t \in I_k, k \in p \qquad \Rightarrow \qquad Y_t \sim \mathcal{N}(m_p, \sigma^2)$$
$$Y^k | Z_{kp} = 1 \sim \mathcal{N}(m_p, \sigma^2).$$

- It is a model of segmentation / clustering.
- Model parameters are $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\gamma})$ and the breakpoint positions $\mathbf{T} = (t_1, ..., t_{K-1})$.

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) (γ, 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 $\boldsymbol{\theta}$;

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

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

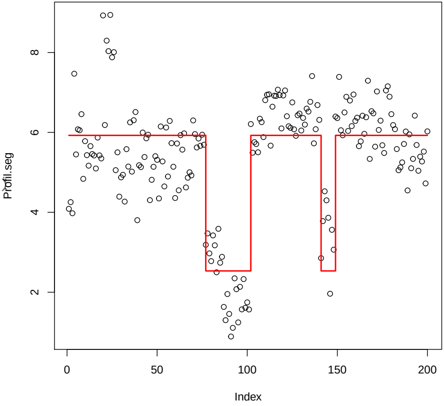
2. When $\boldsymbol{\theta}$ is fixed, dynamic programming estimates $\mathbf{T}$;

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

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

How to perform this segmentation/clustering approach ?

```
## format data into CGHdata
calling(CGHo)<- TRUE ## no classification
CGHo@nblevels=2
segSignal   <- uniseg(.Object=signalCGH,CGHo=CGHo)
segSignalProf <- getsegprofiles(segSignal)
plot(Profil.seg)
lines(1:length(segSignalProf),
      segSignalProf, type="s", col=2, lwd=2)
```



Do it yourself

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 **Z** is a Markov chain if

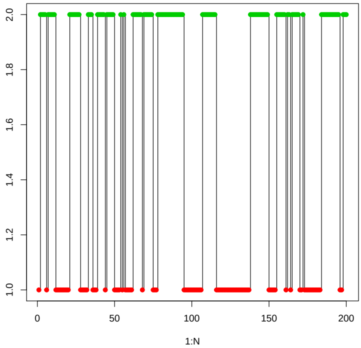
$$P(Z_{i+1} | Z_{1:i}) = P(Z_{i+1} | Z_i)$$

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

```
### 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 $\mathbf{Z}$: $(Y_i|Z_i = k) \stackrel{i.i.d}{\sim} f_{\gamma_k}()$.

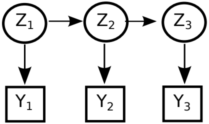
Model parameters are $\boldsymbol{\theta} = (\boldsymbol{\nu}, \mathbf{\Pi}, \gamma)$

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

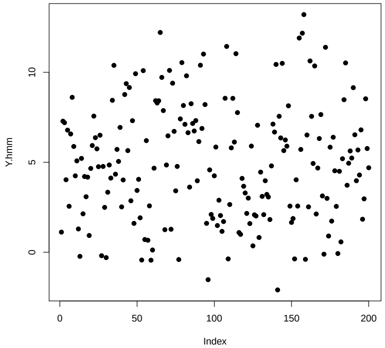


Hidden Markov Chain simulation

```
### observation simulation
mu <- c(3, 7)
sigma <- c(2,3)
Y.hmm <- rnorm(N, mean=mu[Z], sd=sigma[Z])
plot(Y.hmm, pch=19)
plot(Y.hmm, pch=19, col=Z+1)
```

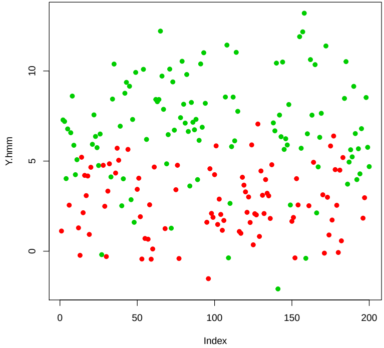
Hidden Markov Chain simulation

```r
### observation simulation
mu <- c(3, 7)
sigma <- c(2,3)
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(2,3)
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

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

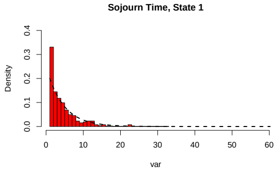
```
switchTime <- which(diff(Z)!=0)
sojournTime <- diff(switchTime)
sojournState <- rep(c(3-Z[1], Z[1]),
                    length.out = length(sojournTime))
br <- unique(quantile(sojournTime,
                      p<- seq(1/N, 1, length.out = 6)))

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)
})
```

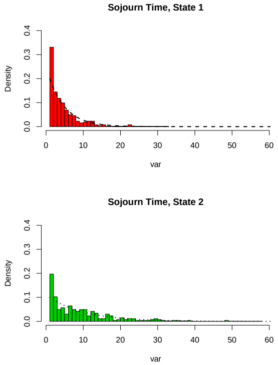


Sojourn time properties

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)
})
```

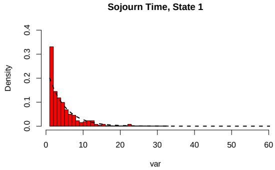


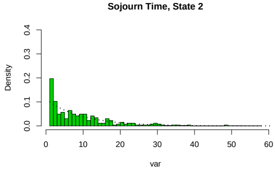
Sojourn time properties

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

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

```
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

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

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

Statistical inference of incomplete data models

Maximum likelihood estimate: We are looking for

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

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

Statistical inference of incomplete data models

Maximum likelihood estimate: We are looking for

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

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

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

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

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

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

EM Algorithm (Baum Welch)

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

  E-step Smart algorithm Forward-Backward algorithm

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

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

Reconstruction of hidden state $\mathbf{Z}$

Most credible value for Z_i: We are interested in

$$\underset{k}{\operatorname{argmax}} \, \mathbb{P}(Z_i = k|\mathbf{Y}) = \underset{k}{\operatorname{argmax}} \, \tau_{ik}.$$

Reconstruction of hidden state **Z**

Most credible value for Z_i: We are interested in

$$\underset{k}{\operatorname{argmax}}\, \mathbb{P}(Z_i = k|\mathbf{Y}) = \underset{k}{\operatorname{argmax}}\, \tau_{ik}.$$

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

$$\underset{k_1,\ldots k_n}{\operatorname{argmax}}\, \mathbb{P}(Z_1 = k_1, \ldots Z_n = k_n|\mathbf{Y}) = ???$$

Reconstruction of hidden state **Z**

Most credible value for Z_i: We are interested in

$$\operatorname*{argmax}_k \mathbb{P}(Z_i = k|\mathbf{Y}) = \operatorname*{argmax}_k \tau_{ik}.$$

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

$$\operatorname*{argmax}_{k_1,\dots k_n} \mathbb{P}(Z_1 = k_1, \dots Z_n = k_n|\mathbf{Y}) = ???$$

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)
```

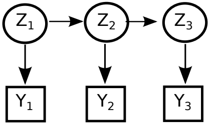
State Space model

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

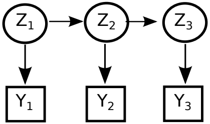


Z could be thougth as the actual locations and **Y** the observed locations.

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.