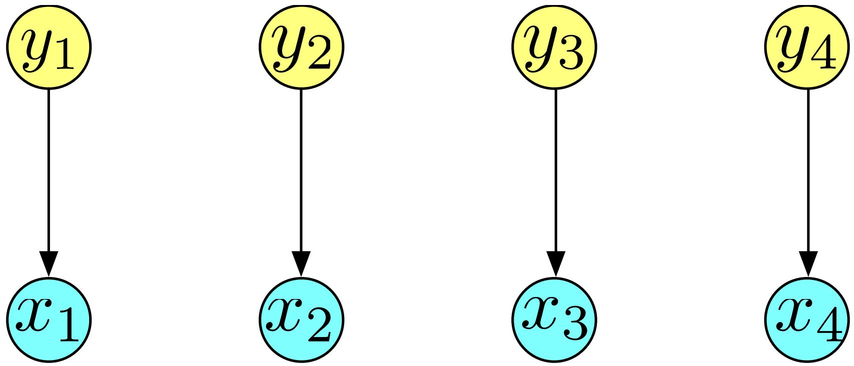
« BAYESIAN LEARNING »

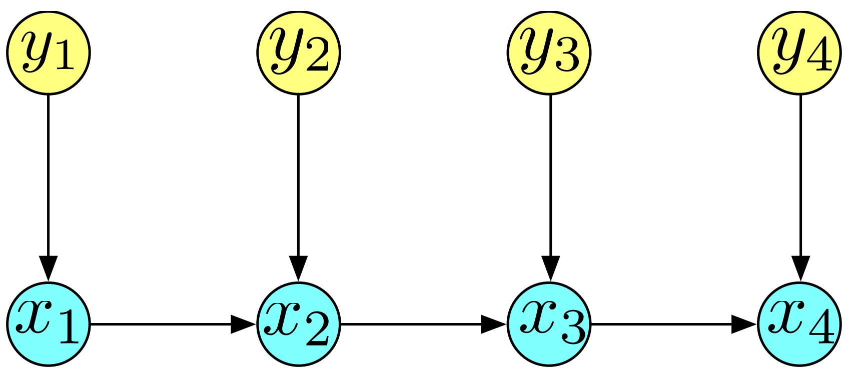
3. HMC MODEL

Stéphane Derrode, Dpt MI -
Stephane.derrode@ec-lyon.fr

Introduction



Mixture model



Markov chain

Hidden Markov chain

Contents for *3. HMC model*

1. Markov chain model
2. Hidden Markov chain model
3. Bayesian decision
4. Unsupervised parameters learning → practical work

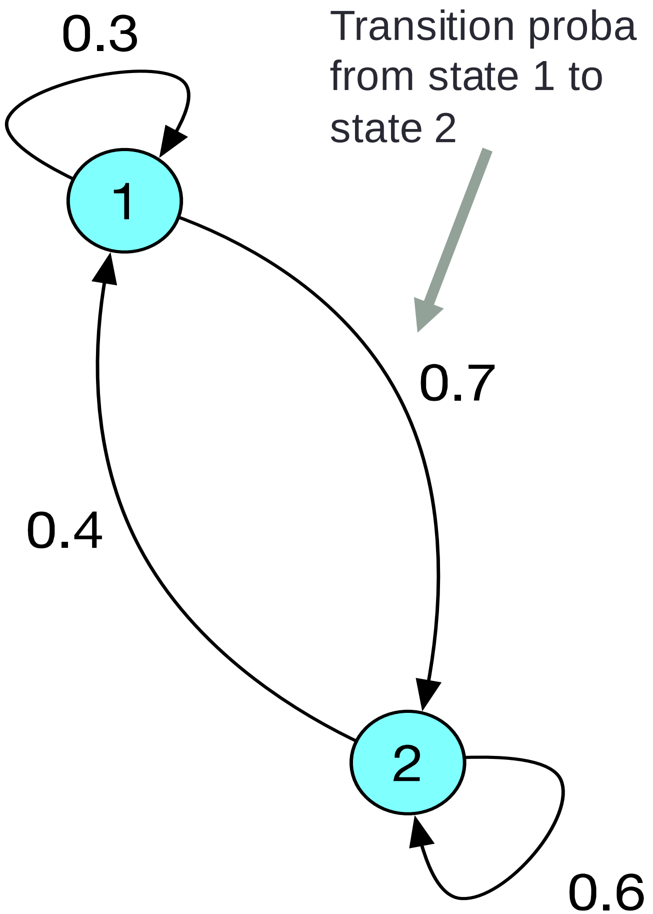
Andreï Markov
1856-1922

HMC MODEL

Markov chain model

Markov chain à temps et état discrets

A **Markov chain** is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event.

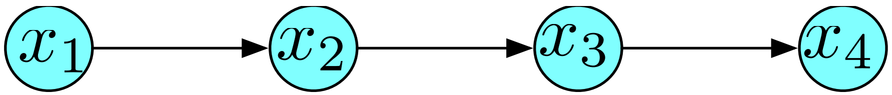


Online accessible book: S. P. Meyn and R.L. Tweedie, 2005. [Markov Chains and Stochastic Stability]

A stochastic process has the **Markov property** if the conditional probability distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it.

$$p(X_n = x_n | X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = p(X_n = x_n | X_{n-1} = x_{n-1})$$

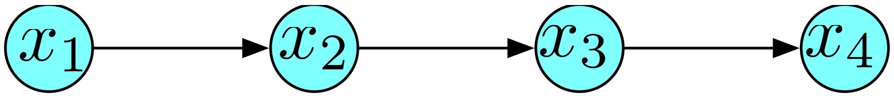
(given the current state, the past doesn't matter)

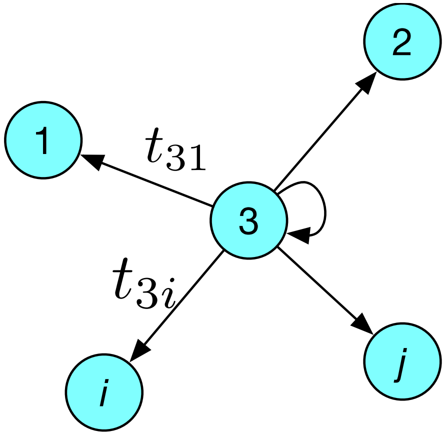


- *Xn* (state after *n* transitions)
  - belongs to a finite set $\Omega = \{1, \ldots, K\}$
  - X_0 is either given or random

- Stationnary Markov transition matrix

$$t_{ij} = p\left(X_{n+1} = j | X_n = i\right)$$





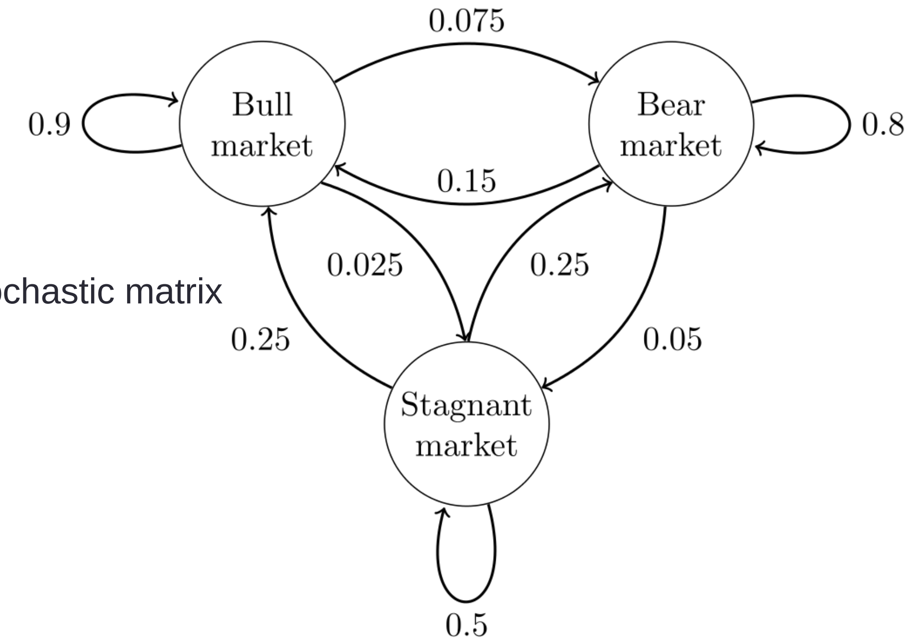
The distribution of a homogeneous Markov chain is entirely determined by its initial distribution and its transition matrix.

Example of state diagram

Labelling the state-space {1 = bull, 2 = bear, 3 = stagnant}, the transition matrix for this example is

$$t = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

Stochastic matrix



The distribution over states can be written as a stochastic row vector X with the relation $X^{(n+1)} = X^{(n)} t$.

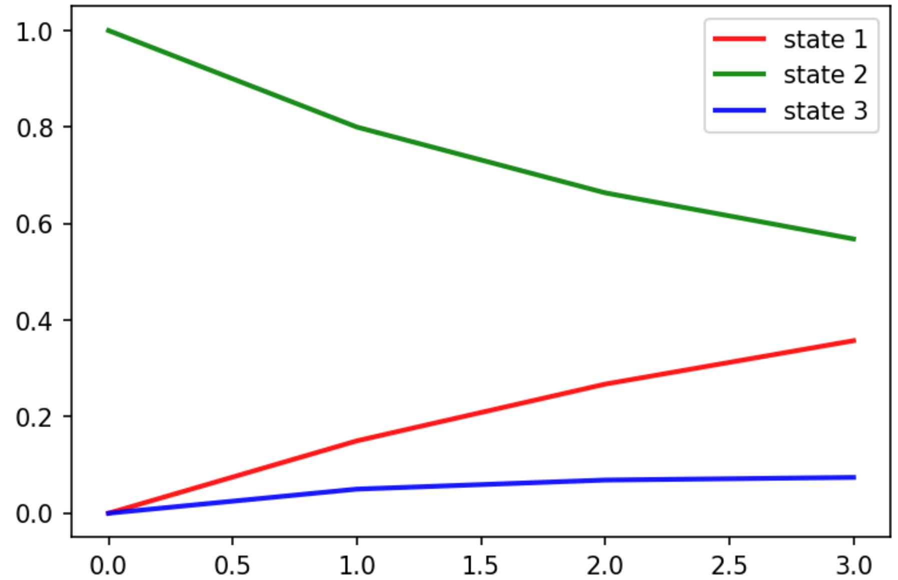
https://en.wikipedia.org/wiki/Markov_chain

Example of state diagram

$$x^{(n+3)} = x^{(n)}\ t^3$$

$$t = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

$$I = x^{(0)} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$



$$x^{(3)} = x^{(0)}\ t^3 = \begin{bmatrix} 0.3575 & 0.56825 & 0.07425 \end{bmatrix}$$

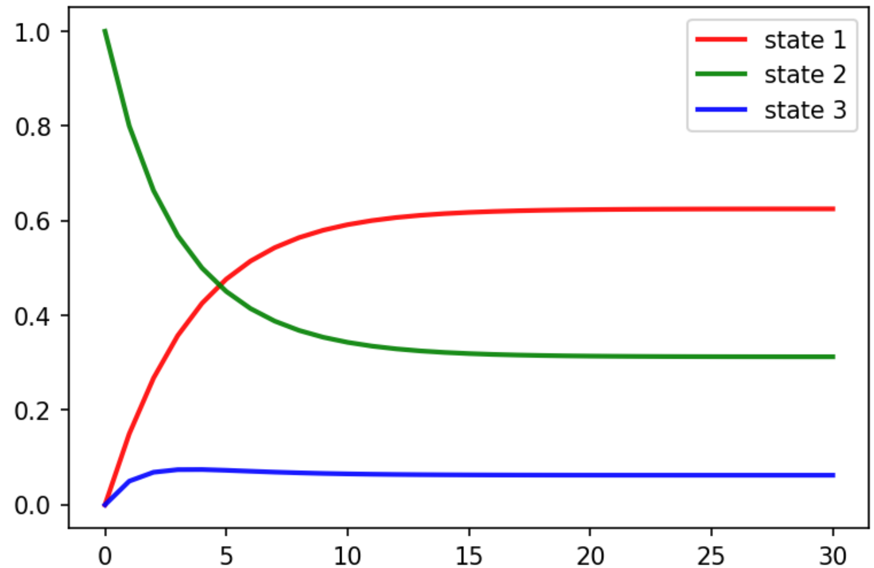
Program: MarkovChain.py

Example of state diagram

$$t = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

$$I = x^{(0)} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

Steady-state distribution



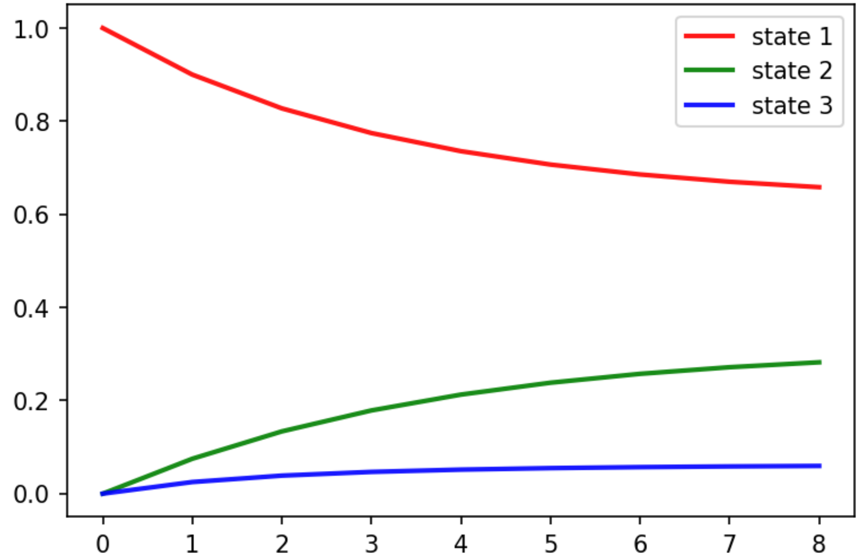
$$x^{(30)} = \begin{bmatrix} 0.62491577 & 0.312577 & 0.06250723 \end{bmatrix}$$

Program: MarkovChain.py

Example of state diagram

$$t = \begin{bmatrix} 0.9 & 0.075 & 0.025 \\ 0.15 & 0.8 & 0.05 \\ 0.25 & 0.25 & 0.5 \end{bmatrix}$$

$$I = x^{(0)} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$



Steady-state distribution (independent from the starting state)

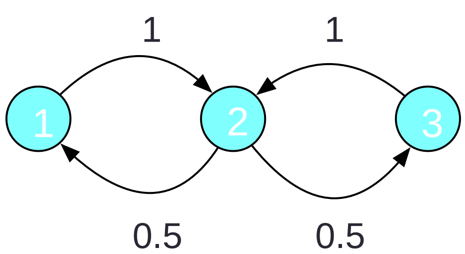
$$x^{(30)} = \begin{bmatrix} 0.62491577 & 0.312577 & 0.06250723 \end{bmatrix}$$

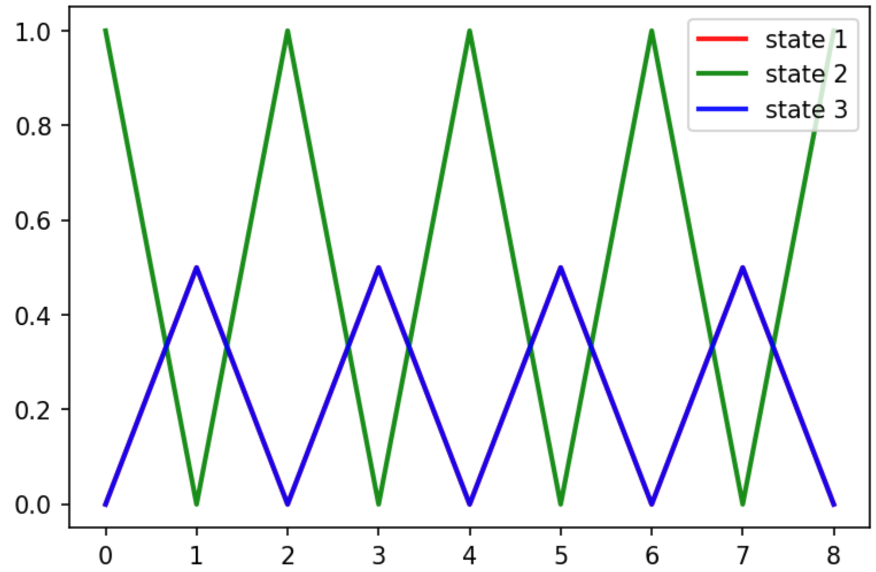
Program: MarkovChain.py

Example of state diagram

$$t = \begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 0.5 & 0.0 & 0.5 \\ 0.0 & 1.0 & 0.0 \end{bmatrix}$$

$$I = x^{(0)} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

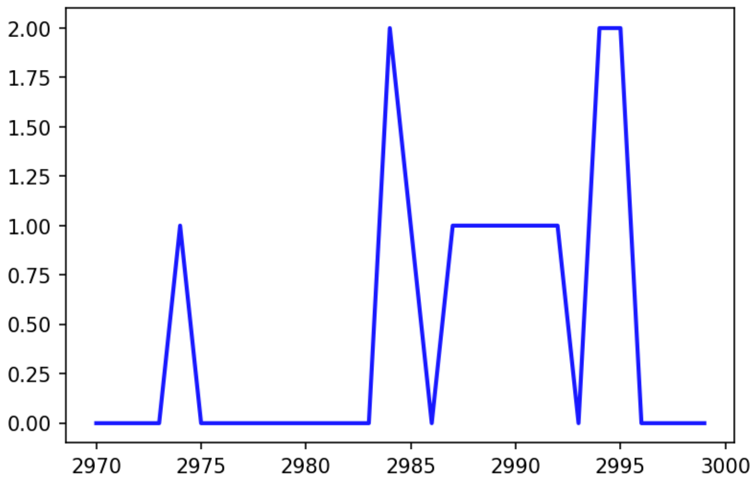




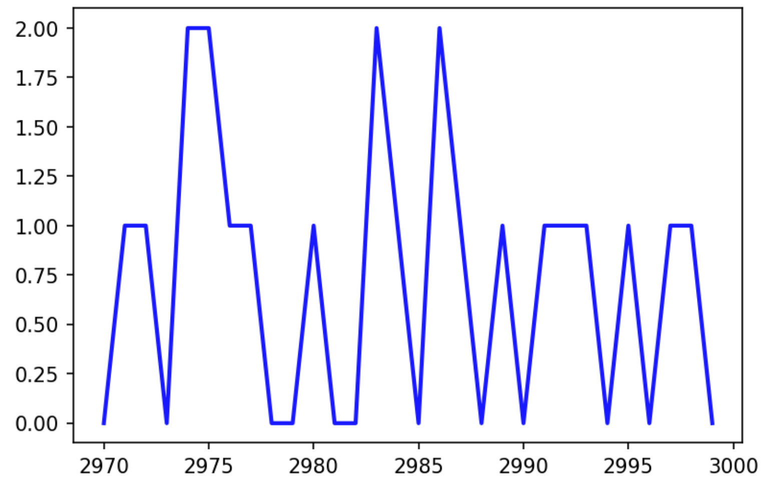
State diagram is 2-periodic
(no Steady-state distribution)

Program: MarkovChain.py

$\hat{\pi} = \begin{bmatrix} 0.46 & 0.30 & 0.24 \end{bmatrix}$



$\hat{\pi} = \begin{bmatrix} 0.31 & 0.39 & 0.30 \end{bmatrix}$



$$I = \begin{bmatrix} 0.1 & 0.55 & 0.35 \end{bmatrix}$$

$$t = \begin{bmatrix} 0.7 & 0.1 & 0.2 \\ 0.3 & 0.6 & 0.1 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$$

$$t = \begin{bmatrix} 0.2 & 0.4 & 0.4 \\ 0.4 & 0.3 & 0.3 \\ 0.3 & 0.5 & 0.2 \end{bmatrix}$$

Program: SimulMarkovChain.py

HMC MODEL

Hidden Markov chain model

A HMC $(\boldsymbol{X}, \boldsymbol{Y})$ is then defined by:

1. A discrete-time homogeneous stationary irreducible Markov chain :

$$c_{ij} = p\left(X_0 = i, X_1 = j\right) = p\left(X_n = i, X_{n+1} = j\right)$$

with initial and stationary law

$$\pi_i = p\left(X_n = i\right) = \sum_{j=1}^{K} c_{ij}$$

and transitions

$$t_{ij} = \frac{c_{ij}}{\pi_i}$$

The distribution of the MC can be written

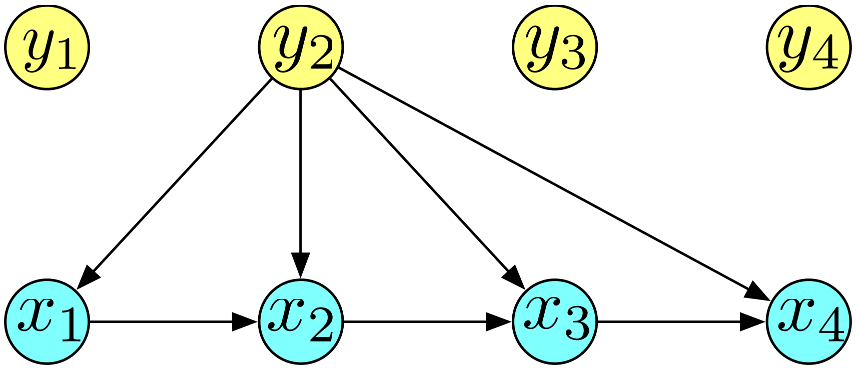
$$p\left(\boldsymbol{X}\right) = \pi_{x_1} \prod_{n=2}^{N} t_{x_{n-1} x_n}$$

A HMC is then defined by:

2. A discrete time "observed" process $\boldsymbol{Y} = \{Y_1, \ldots, Y_N\}$ such that

H1: Random variables Y_n are independent conditionally to $\boldsymbol{X}$

$$p\left(\boldsymbol{Y}|\boldsymbol{X}\right) = \prod_{n=1}^{N} p\left(y_n|\boldsymbol{X}\right)$$

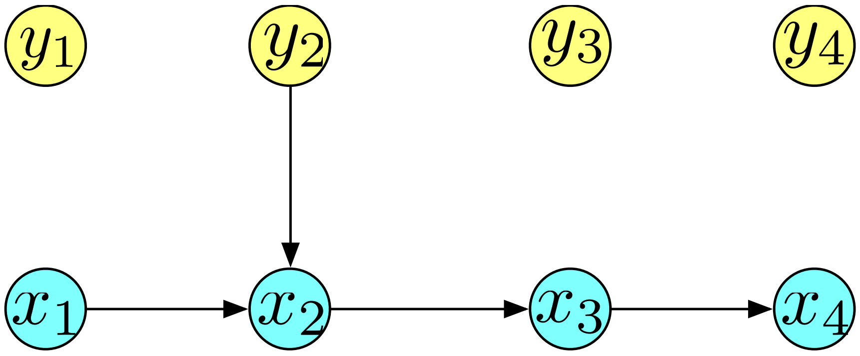


A HMC is then defined by:

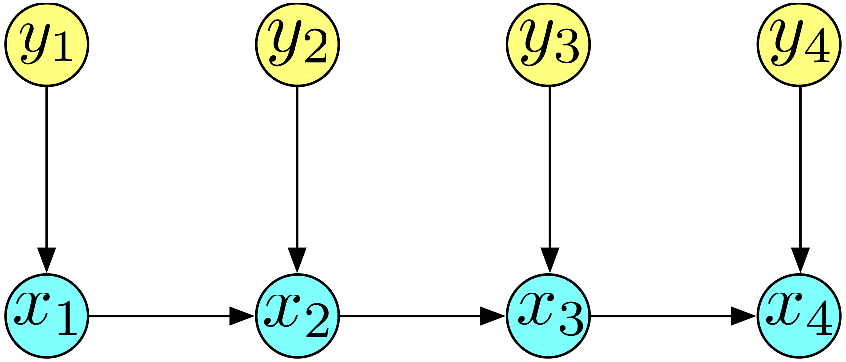
2. A discrete time “observed” process $\boldsymbol{Y} = \{Y_1, \ldots, Y_N\}$ such that

H2: The distribution of each Y_n conditionally to $\boldsymbol{X}$ is equal to its distribution conditionally to X_n

$$
\begin{aligned}
p\left(Y_n | \boldsymbol{X}\right) &= p\left(Y_n | X_n\right) \\
&= f_{x_n}(y_n)
\end{aligned}
$$



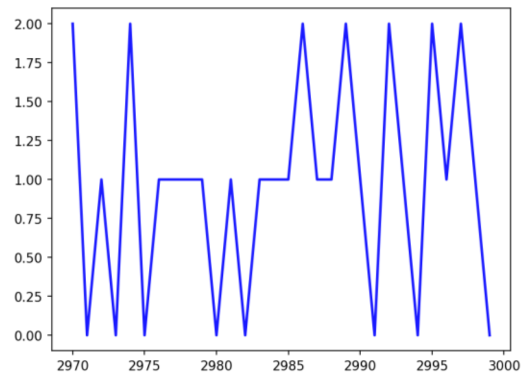
Hidden Markov chain



$$p\left(\boldsymbol{X},\boldsymbol{Y}\right)=\pi_{x_1}f_{x_1}\left(y_1\right)\prod_{n=2}^{N}t_{x_{n-1},x_n}\,f_{x_n}\left(y_n\right)$$

Simulations:

1. Sample a MC of size *N*



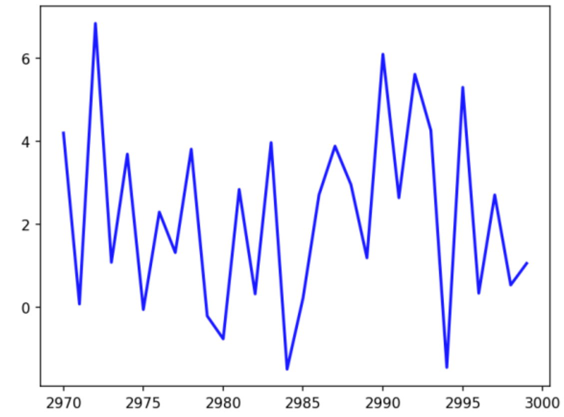
2. Sample observation (conditionally to states)

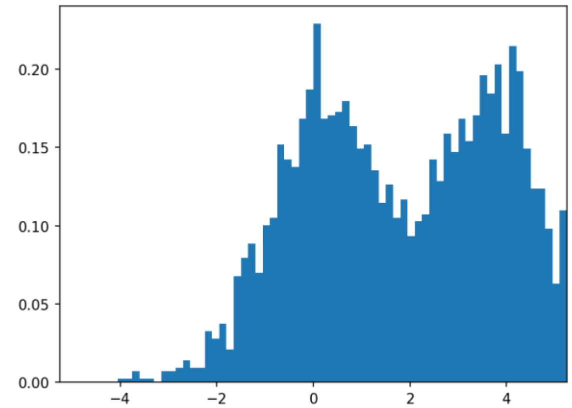
$$t = \begin{bmatrix} 0.7 & 0.1 & 0.2 \\ 0.3 & 0.6 & 0.1 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$$

$$I = \begin{bmatrix} 0.1 & 0.55 & 0.35 \end{bmatrix}$$

$$\mu = \begin{bmatrix} 0 & 2 & 4 \end{bmatrix}$$

$$\sigma = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$





$$\hat{\pi} = \begin{bmatrix} 0.46 & 0.30 & 0.24 \end{bmatrix}$$

Program: SimulHiddenMarkovChain.py

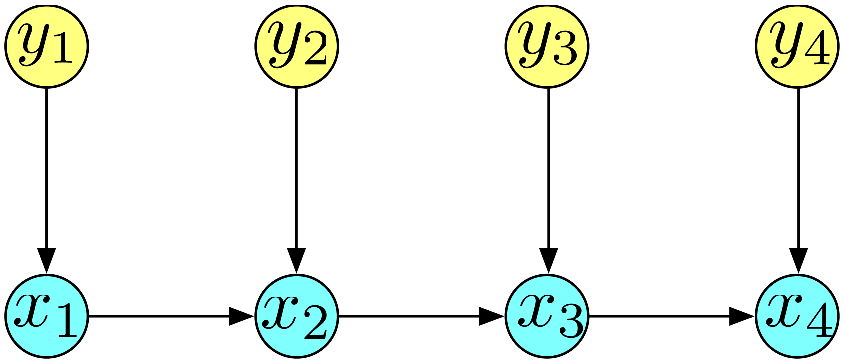
HMC MODEL

Bayesian decision

Bayesian decision

$$p\left(X_n = i, X_{n+1} = j\right) = c_{ij}$$

$$p\left(Y_n | X_n = j\right) \rightsquigarrow \mathcal{N}\left(\mu_j, \sigma_j^2\right)$$



Two questions:

- How to classify according to the Bayesian Decision criterion (assuming known parameters)?
- How to learn parameters automatically?

Bayesian decision: minimize the mean error rate of classification.

In the independent data case, assuming the loss function

$$L(i,j) = \begin{cases} 0 & \text{if } i = j \\ \lambda_{i,j} > 0 & \text{else} \end{cases}$$

the Bayesian decision can be written

$$\hat{s}_B(y) = k = \arg\min_{j \in \Omega} \sum_{i=1}^{K} \lambda_{j,i} \, p(X = i | Y = y)$$

In a general fashion

$$\begin{aligned} \hat{s}_B(\boldsymbol{y}) &= \arg\min_{\hat{\boldsymbol{x}} \in \Omega^N} E\left[L\left(\boldsymbol{X} = \boldsymbol{x}, \hat{\boldsymbol{x}}\right) | \boldsymbol{Y} = \boldsymbol{y}\right] \\ &= \arg\min_{\hat{\boldsymbol{x}} \in \Omega^N} \sum_{\boldsymbol{x} \in \Omega^N} L\left(\boldsymbol{x}, \hat{\boldsymbol{x}}\right) \, p\left(\boldsymbol{x} | \boldsymbol{y}\right) \end{aligned}$$

For HMC, we consider two criterions (*ie.* two loss functions):

- MPM: Marginal Posterior Mode
- MAP: Maximum a posteriori

MPM criterion

Loss function $L_1(\hat{\boldsymbol{x}}, \boldsymbol{x}) = \sum_{n=0}^{N} L(x_n, \hat{x}_n)$

MPM (Marginal Posterior Mode) estimator

$$\forall n \in [0, \ldots, N], \quad \hat{x}_n^{MPM}(\boldsymbol{y}) = \arg\max_{k \in \Omega} \left[p\left(X_n = k | \boldsymbol{Y}\right) = \tilde{\gamma}_n(k) \right]$$

Posterior marginal proba of the states given all the observations

$$\tilde{\gamma}_n(k) = p(X_n = k | \boldsymbol{Y}) = \alpha_n(k)\ \beta_n(k)$$

Smoothing probabilities Forward (filtering) probabilities Backward probabilities

Forward probabilities

$$\alpha_n(k) = p(X_n = k|\boldsymbol{y}_0^n) = \frac{1}{S_n}\, p\left(X_n = k, y_n|\boldsymbol{y}_0^{n-1}\right)$$

with $S_n = p\left(y_n|\boldsymbol{y}_0^{n-1}\right), n > 0$

n=0 $\forall k \in \Omega, \alpha_0(k) = p(X_0 = k|y_0) = \frac{p(X_0 = k, y_0)}{p(y_0)} = \frac{\pi_k f_k(y_0)}{\sum_{l=1}^K \pi_l f_l(y_0)}$

n>0 $\forall k \in \Omega, \alpha_{n+1}(k) = \frac{1}{S_{n+1}}\, f_k(y_{n+1}) \sum_{l=1}^{K} t_{lk}\, \alpha_n(l)$

Backward probabilities

$$\beta_n(k) = \frac{p\left(\boldsymbol{y}_{n+1}^N | X_n = k\right)}{p\left(\boldsymbol{y}_{n+1}^N | \boldsymbol{y}_1^n\right)} = \frac{1}{S_{n+1}} \frac{p\left(\boldsymbol{y}_{n+1}^N | X_n = k\right)}{p\left(\boldsymbol{y}_{n+2}^N | \boldsymbol{y}_1^{n+1}\right)}$$

avec $S_n = p\left(y_n | y_0^{n-1}\right), n > 0$

n=N $\forall k \in \Omega, \beta_N(k) = 1$

n<N $\forall k \in \Omega, \beta_n(k) = \frac{1}{S_{n+1}} \sum_{l=1}^{K} t_{kl} \; f_l(y_{n+1}) \; \beta_{n+1}(l)$

Posterior marginal proba of the states given all the observations

$$\tilde{\gamma}_n(k) = p(X_n = k|\boldsymbol{Y}) = \alpha_n(k)\ \beta_n(k)$$

MPM (Marginal Posterior Mode) estimator

$$\forall n \in [0, \ldots, N], \quad \hat{x}_n^{MPM}(\boldsymbol{y}) = \arg\max_{k \in \Omega} \left(\tilde{\gamma}_n(k)\right)$$

MAP Criterion

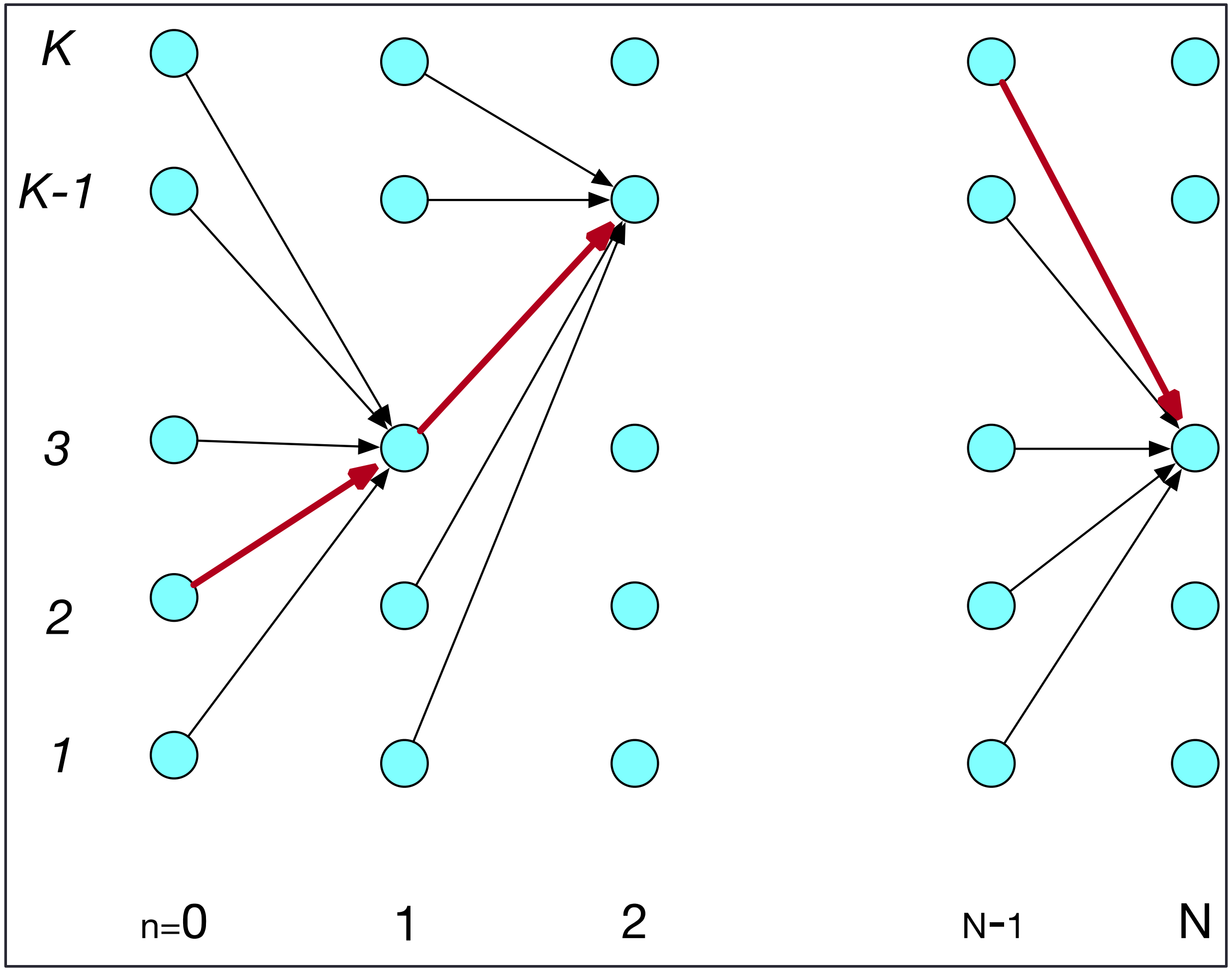
Loss function

$$L_2(\hat{\boldsymbol{x}}, \boldsymbol{x}) = \mathbf{1}_{\{\hat{\boldsymbol{x}} \neq \boldsymbol{x}\}}$$

MAP (Maximum A Posteriori) estimator

$$\begin{aligned}
\hat{\boldsymbol{x}}^{MAP}(\boldsymbol{y}) &= \arg\max_{\boldsymbol{x} \in \Omega^N} p\left(\boldsymbol{X} = \boldsymbol{x} | \boldsymbol{y}\right) \\
&= \arg\max_{\boldsymbol{x} \in \Omega^N} p\left(\boldsymbol{x}, \boldsymbol{y}\right) \\
&= \arg\max_{\boldsymbol{x} \in \Omega^N} p\left(\boldsymbol{y} | \boldsymbol{x}\right)\ p\left(\boldsymbol{x}\right)
\end{aligned}$$

Viterbi's algorithm


K
K-1
3
2
1
n=0
1
2
N-1
N

Cost of a 1-transition

$$d(x_{n-1} = i, x_n = j) = p(x_n = j, y_n | x_{n-1} = i, y_{n-1}) = t_{ij}\ f_j(y_n)$$

with initial cost

$$d(x_0 = j) = \pi_j\ f_j(y_0)$$

Total cost for a path c

$$P = \prod_{n=0}^{N} d\left(x_{c(n-1)}, x_{c(n)}\right)$$

We seek to optimize

$$D = \ln(P) = \sum_{n=0}^{N} \ln\left(d\left(x_{c(n-1)}, x_{c(n)}\right)\right)$$

Maximum cost to reach state *k* at *n*

$$\delta_n(k) = \ln\left(f_k(y_n)\right) + \max_{j\in\Omega}\left(\delta_{n-1}(j) + \ln\left(t_{jk}\right)\right)$$

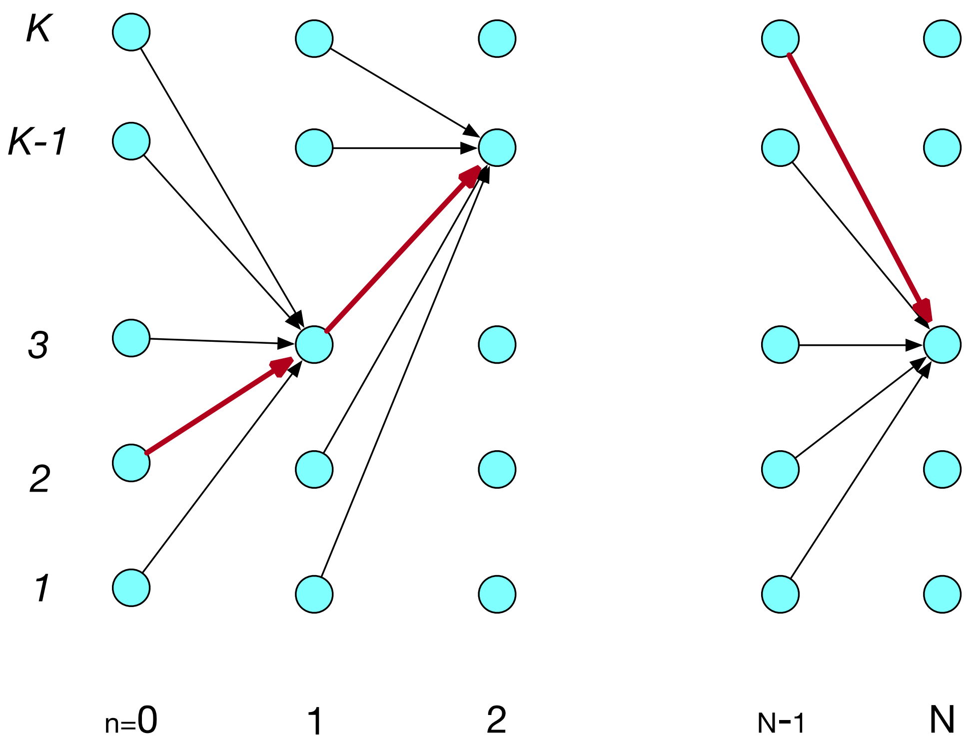
$$\psi_n(k) = \arg\max_{j\in\Omega}\left(\delta_{n-1}(j) + \ln\left(t_{jk}\right)\right)$$

with $\delta_0(k) = \ln(\pi_k) + \ln(f_k(y_0))$

Path reconstruction (backward) - decoding

$$\hat{x}_N^{MAP} = \arg\max_{k\in\Omega} \delta_N(k)$$

$$\hat{x}_{n-1}^{MAP} = \psi_{n+1}(\hat{x}_n^{MAP})$$



HMC MODEL

Unsupervised parameter learning

Unsupervised parameter learning

Law of $\boldsymbol{X}$ a posteriori :

$$\tilde{c}_n(k,l) = p\left(X_n = k, X_{n+1} = l | \boldsymbol{Y}\right) = \frac{\alpha_n(k)\ \beta_{n+1}(l)\ f_l(y_{n+1})\ t_{kl}}{S_{n+1}}$$

$$\tilde{\gamma}_n(k) = p(X_n = k|\boldsymbol{Y}) = \alpha_n(k)\ \beta_n(k)$$

$$\tilde{t}_n(k,l) = p\left(X_{n+1} = k | X_n = k, \boldsymbol{Y}\right) = \frac{\beta_{n+1}(l)\ f_l(y_{n+1})\ t_{kl}}{\beta_n(k)\ S_{n+1}}$$

The chain $\boldsymbol{X}|\boldsymbol{Y}$ is of Markov type, but not homogeneous, and not stationary!

Completed log-likelihood

$$\ln \mathcal{H}_\Theta(\boldsymbol{y}, \boldsymbol{X}) = \ln\left(\pi_{x_0}\right) + \sum_{n=0}^{N} \ln\left(f_{x_n}(y_n)\right) + \sum_{n=1}^{N} \ln t_{x_{n-1}x_n}$$

Auxiliary function

$$\mathcal{Q}\left(\Theta; \Theta^{(\ell)}\right) = \sum_{k=1}^{K} \tilde{\gamma}_0(k) \ln\left(\pi_k^{(\ell)}\right) + \sum_{k=1}^{K}\sum_{n=0}^{N} \tilde{\gamma}_n(k) \ln\left(f_k^{(\ell)}(y_n)\right) + \sum_{k=1}^{K}\sum_{i=1}^{K}\sum_{n=1}^{N} \tilde{c}_n(k,i) \ln\left(t_{ki}^{(\ell)}\right)$$

$$\mu_k^{(\ell+1)} = \frac{\sum_{n=1}^{N} \tilde{\gamma}_n(k)\; y_n}{\sum_{n=1}^{N} \tilde{\gamma}_n(k)}$$

$$\pi_k^{(\ell+1)} = \frac{1}{N} \sum_{n=1}^{N} \tilde{\gamma}_n(k)$$

$$\sigma_k^{2,(\ell+1)} = \frac{\sum_{n=1}^{N} \tilde{\gamma}_n(k) \; \left(y_n - \mu_k^{(\ell+1)}\right)^2}{\sum_{n=1}^{N} \tilde{\gamma}_n(k)}$$

$$t_{ki}^{(\ell+1)} = \frac{\sum_{n=1}^{N-1} \tilde{c}_n(k,i)}{\sum_{n=1}^{N-1} \tilde{\gamma}_n(k)}$$

Algorithm

Require: Le nombre de classes K, et un signal à valeurs réelles $\underline{\boldsymbol{y}}$

1. Initialisation : Donner une valeur initiale aux paramètres.

Segmenter $\underline{\boldsymbol{y}} \longrightarrow \underline{\boldsymbol{x}}^{(0)}$

Estimer les paramètres sur les données complètes $\left(\underline{\boldsymbol{y}}, \underline{\boldsymbol{x}}^{(0)}\right) \longrightarrow \underline{\boldsymbol{\Theta}}^{(0)}$

2. Estimation EM : Trouver $\underline{\boldsymbol{\Theta}}^{(L)}$

for $\ell = 1$ to L **do**

À partir des paramètres de l'itération précédente $\underline{\boldsymbol{\Theta}}^{(\ell)}$

Calculer les probabilités « avant-arrière » : $\alpha_n(k)$ et $\beta_n^{(}k)$

Calculer les probabilités *a posteriori* : $\tilde{\gamma}_n(k)$ et $\tilde{c}_n^{(}k)$

Estimer les paramètres de bruit : $\mu_k^{(\ell+1)}$ et $\sigma_k^{2,(\ell+1)}$.

Estimer les paramètres de Markov : $\pi_k^{(\ell+1)}$ et $t_{ki}^{(\ell+1)}$

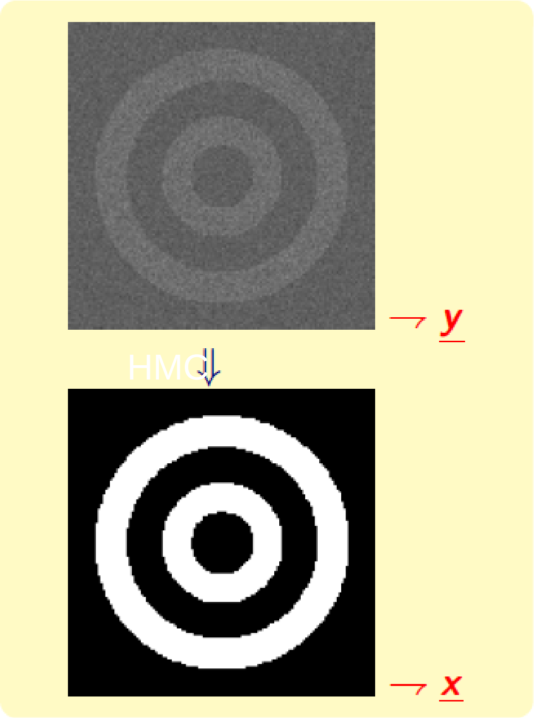
end for

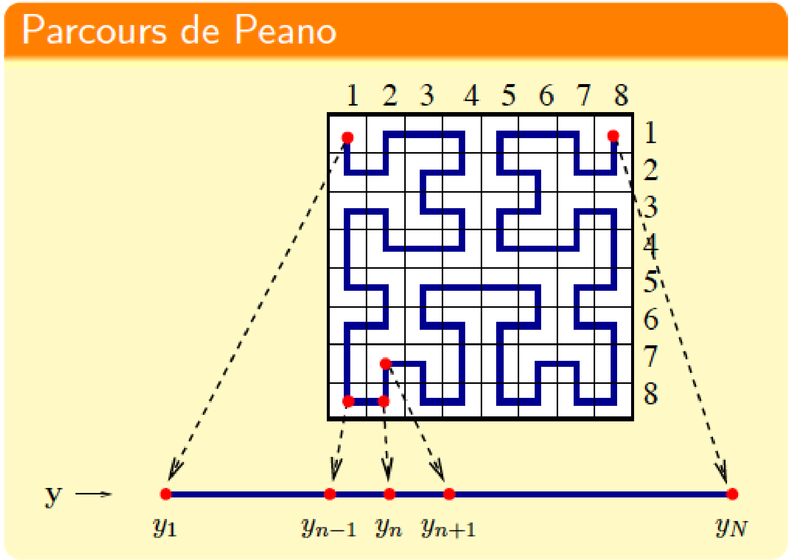
3. Segmentation : Appliquer une décision bayésienne

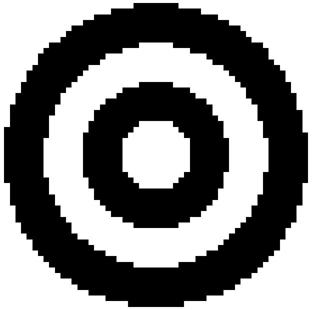
MPM : directement à partir de $\tilde{\gamma}_n(k)$.

MAP : algorithme de Viterbi (utilisant $\boldsymbol{\Theta}^{(L)}$).

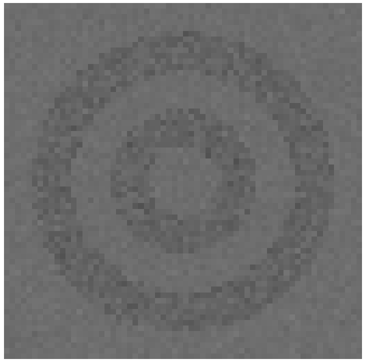
Application to image segmentation



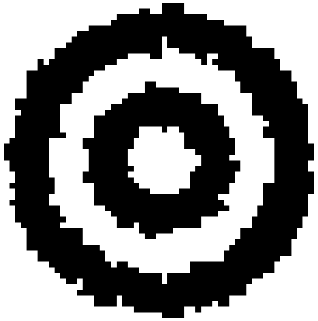




Original image



Noisy image (with MM)



Restored image (MPM criterion)

Final estimations (after 30 EM iterations):

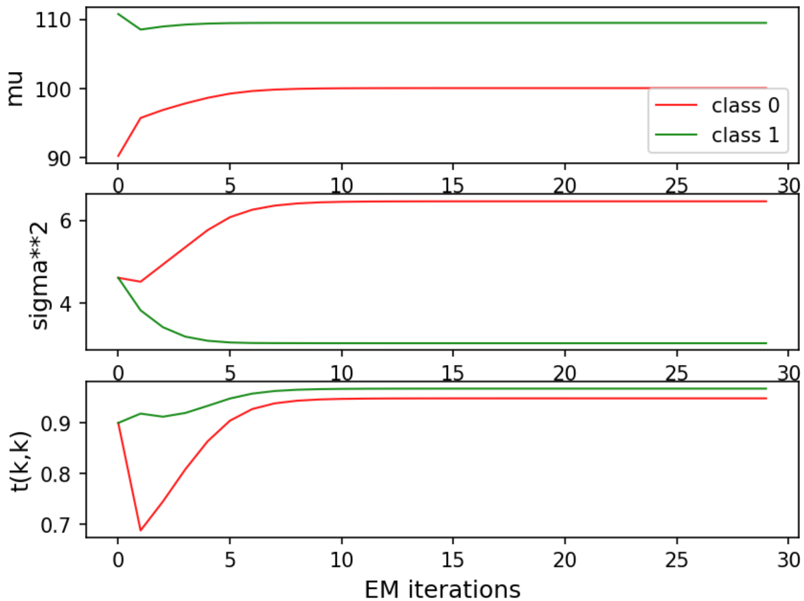
Confusion matrix for MPM =

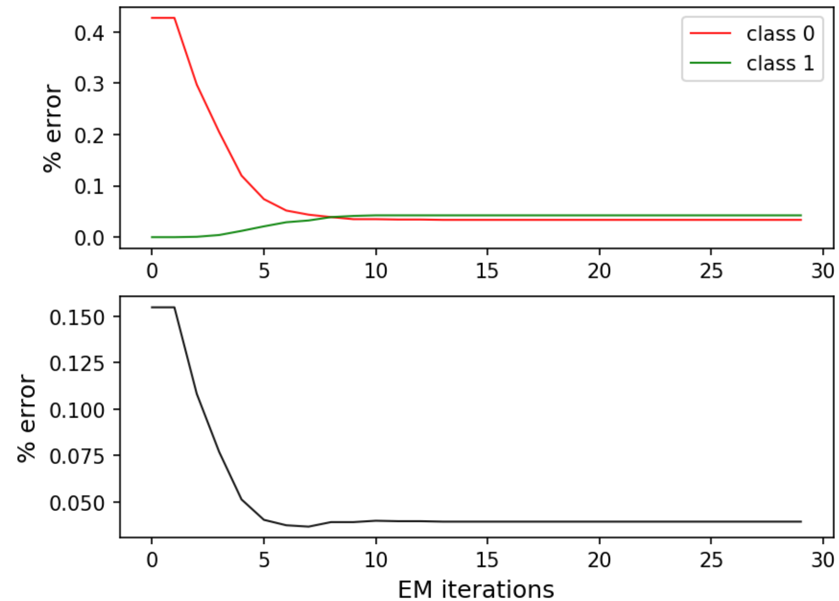
[1434.   50.]

[111. 2501.]

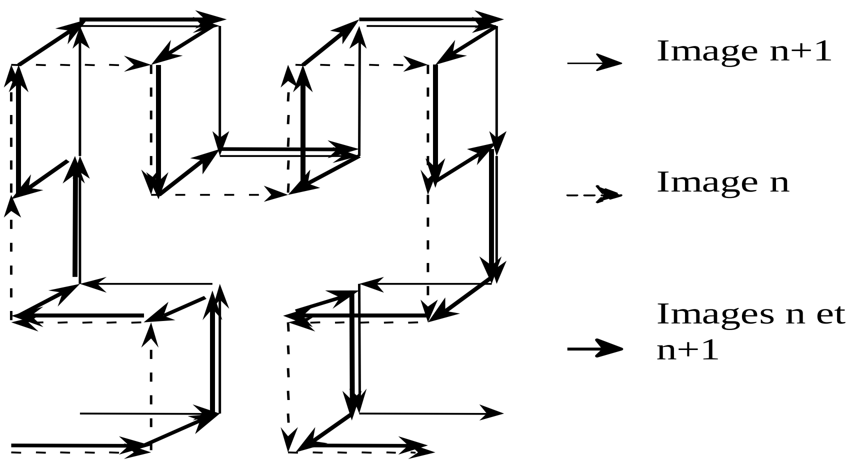
Global Error rate for MPM: 0.039306640625

Class Error rate for MPM: [0.03369272 0.04249617]


mu
sigma**2
t(k,k)
class 0
class 1
EM iterations


% error
class 0
class 1
% error
EM iterations

Parcours 3D (sequence d'images)



Original image

Segmentation from the current image only

Segmentation from the past and current images

Extensions : 30 years of HMC

- EM -> Iterated Conditional Estimation
  - W. Pieczynski, Champs de Markov caché et estimation conditionnelle itérative, *Traitement du Signal*, Vol. 11, No. 2, pp. 141-153, 1994.
- Generalized mixture (non Gaussian – Pearson' system of distributions)
  - S. Derrode and G. Mercier, *Unsupervised multiscale oil slick segmentation from SAR image using a vector HMC model*, *Pattern Recognition*, Vol. 40(3), pp. 1135-1147, 2007.
- Fuzzy Markov chain
  - C. Carincotte, S. Derrode and S. Bourennane, *Unsupervised change detection on SAR images using fuzzy hidden Markov chains*, *IEEE Trans. on Geoscience and Remote Sensing*, Vol. 44(2), pp. 432-441, 2006.
- Pairwise and triplet Markov chain
  - S. Derrode and W. Pieczynski, *Unsupervised signal and image segmentation using pairwise Markov chains*, *IEEE Trans. on Signal Processing*, Vol. 52(9), pp. 2477-2489, 2004.
  - W. Pieczynski, Chaîne de Markov triplet, Triplet Markov Chains, *Comptes Rendus de l'Académie des Sciences – Mathématique*, Série I, Vol. 335, No. 3, pp. 275-278, 2002.