Supervised machine learning
Connecting local models
The case of chains

Romain Raveaux

romain.raveaux@univ-tours.fr

Maître de conférences

Université de Tours

Laboratoire d’informatique (LIFAT)

Equipe RFAI

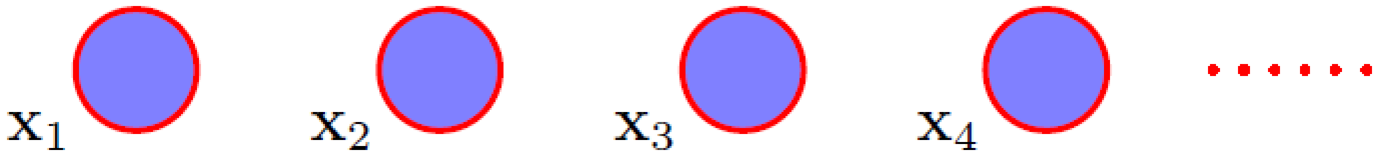
Content :

- This presentation is a follow up to « Supervised machine learning : the case of independent samples »
  - http://romain.raveaux.free.fr/document/courssupervisedmachinelearningRaveaux.pdf
- The case of a chain :
  - Probabilistic models connected to form a chain:
    - A first step beyond Independent and Identically Distributed (I.I.D)
  - Models
  - Inference
  - Learning
  - Directed and undirected models

Independent and Identically Distributed

- A set of samples : $\{x_i\}_{i=1}^{M} = x_1, \cdots, x_M$
  - This set is described by a single distribution. Pr(x)
    - Each sample is drawn from Pr(x)
  - Each sample is independent $Pr(x_i|x_{i-1}) = Pr(x_i)$
  - The joint distribution $\mathrm{Pr}(x_1, \ldots, x_M)$ is the product over all data points of the probability distribution evaluated at each data point.

$$Pr(x_1, \cdots, x_M) = \prod_{i=1}^{M} Pr(x_i)$$



Beyond : Independent and Identically Distributed

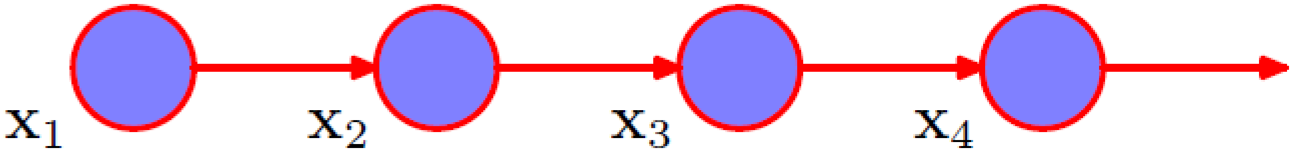
- For many applications, however, the i.i.d. assumption will be a poor one.
  - Describe sequential data (time series)
    - the rainfall measurements on successive days at a particular location
    - the sequence of characters in an English sentence
    - the daily values of a currency exchange rate

$$Pr(x_1, \cdots, x_M) = \prod_{i=1}^{M} Pr(x_i | x_1, \cdots, x_{M-1})$$

A first-order Markov chain

- Markov Assumption :
  - The future depends only on the present.

$$Pr(x_1, \cdots, x_M) = Pr(x_1) \prod_{i=2}^{M} Pr(x_i|x_{i-1})$$



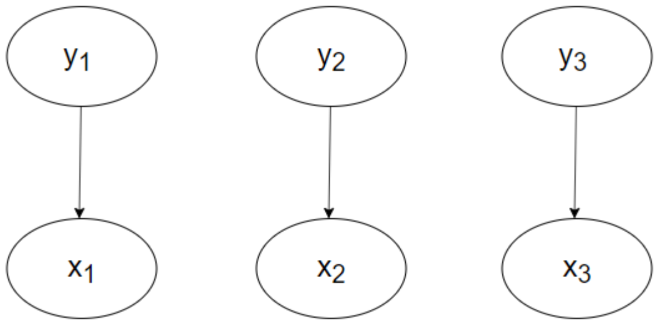
$$Pr(x_i|x_1, \cdots, x_{M-1}) = Pr(x_i|x_{i-1})$$

Ok let’s go back on a generative model.

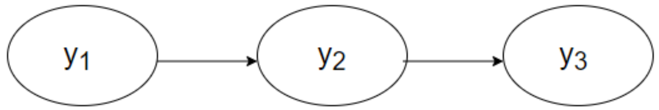
- The product rule gives us : Pr(Y,X)= Pr(X|Y)Pr(Y)

$$Pr(y_1, \cdots, y_M; x_1, \cdots, x_M) = Pr(x_1, \cdots, x_M | y_1, \cdots, y_M).Pr(y_1, \cdots, y_M)$$

$$Pr(x_1, \cdots, x_M | y_1, \cdots, y_M) = \prod_{i=1}^{M} Pr(x_i | y_i)$$



$$Pr(y_1, \cdots, y_M) = Pr(y_1) \prod_{i=2}^{M} Pr(y_i | y_{i-1})$$

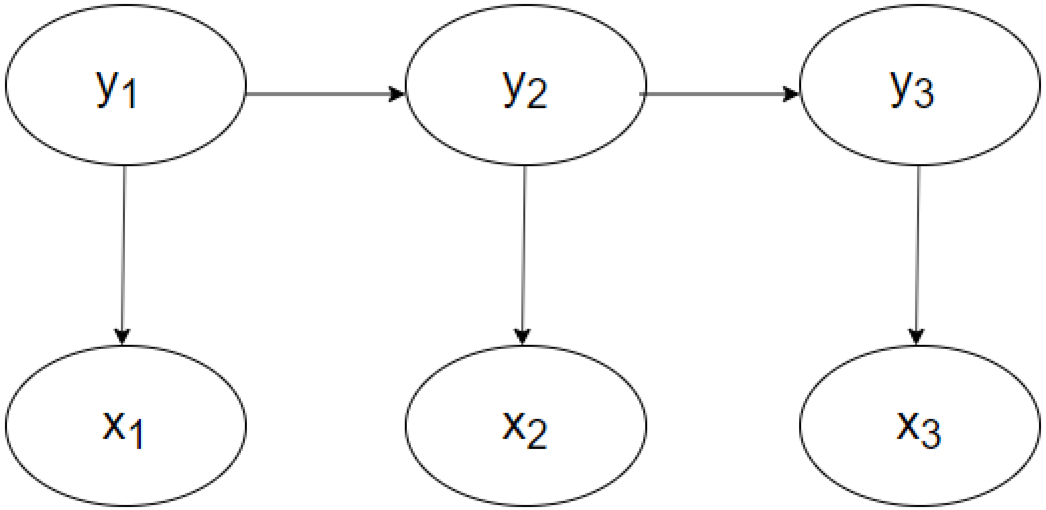


Ok let's go back on a generative model.

- Let's put it together

$$Pr(y_1, \cdots, y_M; x_1, \cdots, x_M) = Pr(x_1, \cdots, x_M | y_1, \cdots, y_M).Pr(y_1, \cdots, y_M)$$

$$Pr(y_1, \cdots, y_M; x_1, \cdots, x_M) = \left[\prod_{i=1}^{M} Pr(x_i|y_i)\right]\left[Pr(y_1)\prod_{i=2}^{M} Pr(y_i|y_{i-1})\right]$$



I know this model

- This is known as a
  - Hidden Markov model (HMM) when y_i is discrete
  - Kalman Filter model when y_i is continuous

Inference for HMM : maximum a posteriori (MAP)

Inference with a generative model:

Use the **baye's rule** to obtain the posterior distribution

$$Pr(y_1, \cdots, y_M | x_1^{new}, \cdots, x_M^{new}) = \frac{Pr(x_1^{new}, \cdots, x_M^{new} | y_1, \cdots, y_M) . Pr(y_1, \cdots, y_M)}{Pr(x_1^{new}, \cdots, x_M^{new})}$$

Inference for HMM

$$Pr(y_1,\cdots,y_M|x_1^{new},\cdots,x_M^{new}) = \frac{Pr(x_1^{new},\cdots,x_M^{new}|y_1,\cdots,y_M).Pr(y_1,\cdots,y_M)}{Pr(x_1^{new},\cdots,x_M^{new})}$$

$$\hat{y_1},\cdots,\hat{y_M} = arg\max_{y_1,\cdots,y_M}\left[Pr(y_1,\cdots,y_M|x_1^{new},\cdots,x_M^{new})\right]$$

$$\hat{y_1},\cdots,\hat{y_M} = arg\max_{y_1,\cdots,y_M}\left[Pr(x_1^{new},\cdots,x_M^{new}|y_1,\cdots,y_M).Pr(y_1,\cdots,y_M)\right]$$

$$\hat{y_1},\cdots,\hat{y_M} = arg\max_{y_1,\cdots,y_M}\left[\prod_{i=1}^{M}Pr(x_i^{new}|y_i).Pr(y_1)\prod_{i=2}^{M}Pr(y_i|y_{i-1})\right]$$

$$\hat{y_1},\cdots,\hat{y_M} = arg\min_{y_1,\cdots,y_M} -\log\left[Pr(x_1^{new},\cdots,x_M^{new}|y_1,\cdots,y_M).Pr(y_1,\cdots,y_M)\right]$$

$$\hat{y_1},\cdots,\hat{y_M} = arg\min_{y_1,\cdots,y_M}\left[-\sum_{i=1}^{M}Pr(x_i^{new}|y_i) - Pr(y_1) - \sum_{i=2}^{M}Pr(y_i|y_{i-1})\right]$$

$$\hat{y_1},\cdots,\hat{y_M} = arg\min_{y_1,\cdots,y_M}\left[\sum_{i=1}^{M}U_i(y_i) + \sum_{i=2}^{M}P_i(y_i,y_{i-1})\right]$$

$$U_i(y_i) = -\log[Pr(x_i^{new}|y_i)]$$

$$P_i(y_i,y_{i-1}) = -\log[Pr(y_i|y_{i-1})]$$

Where

- Ui is a **unary** term and depends only on a single variable y_i and
- Pi is a **pairwise** term, depending on two variables y_i andy_{i-1}.

How can we solve this optimization problem ?

- can be solved in polynomial time using the Viterbi algorithm which is an example of dynamic programming.

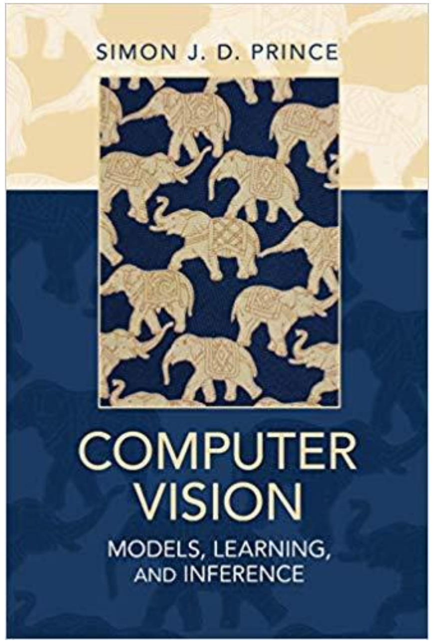
- **We consider that all the distributions are known**

Inference for HMM : MAP

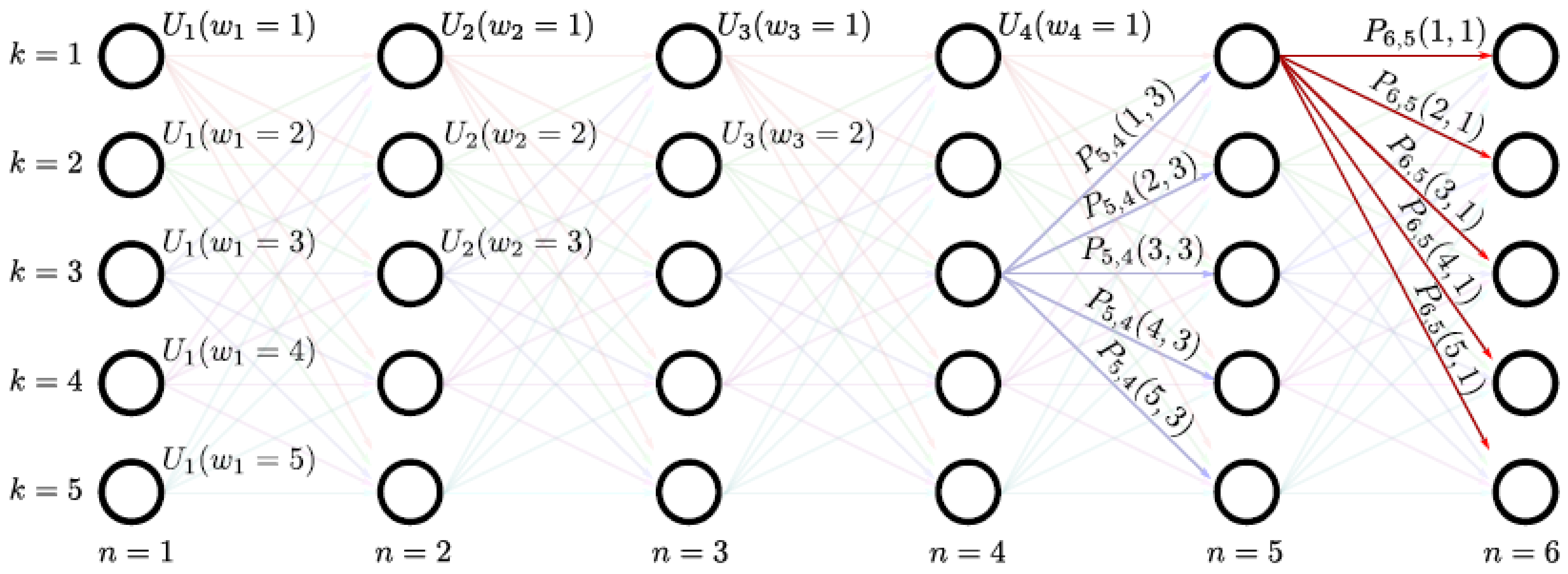
- Case of K classes : $y_i \in [1,2,3,4,5]$
- Build a graph
  - The set of vertices $\{V_{i,k}\}_{i=1,k=1}^{M,5}$
  - Each vertex $V_{i,k}$ has a set of edges $(V_{i-1,l}, V_{i,k})_{l=1}^{5}$
  - Each vertex $V_{i,k}$ has a cost $U_i(y_i = k)$
  - Each edge $(V_{i-1,l}, V_{i,k})$ has a cost $P_i(y_i = k, y_{i-1} = l)$
- Find the shortest path between left to right
  - Where the notion of distance : d($(V_{i-1,l}, V_{i,k})$) = U_{i-1}+P_i
  - Dijkstra can be used

Picure taken from:

Inference for HMM : Viterbi in short



- For this slide only: Change of notation y variables become w
- For this slide only: Change of notation n=i



- Find the shortest path from left to right $S_{1,k} = U_1(w_1 = k)$

$$S_{2,k} = U_2(w_2 = k) + \min_l \left[S_{1,l} + P_2(w_2 = k, w_1 = l)\right] \qquad S_{n,k} = U_n(w_n = k) + \min_l \left[S_{n-1,l} + P_n(w_n = k, w_{n-1} = l)\right]$$

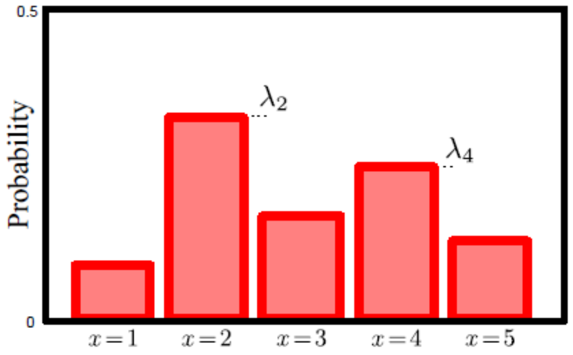
Inference for HMM : Complexity

- MAP inference :
  - Brute force approach : $\mathrm{O}(K^M)$
    - Enumerating all the combinations
      - Sequence of length 3, K=5 : 111; 112 ; 113; 114; 115; 121; 122; ….
  - Viterbi : $\mathrm{O}(\mathrm{M}K^2)$

Learning with HMM

- So far there is no learning.
  - We just give a sequence (x) and output the labeled sequence (y)
- Where learning can be introduced ?
  - Where are the parameters ?
  - The measurements (x) have a normal distribution
  - The class variable (y) follows a categorical law.
  - This hidden Markov model has parameters $\{\mu_k, \sigma_k, \lambda_k\}_{k=1}^{K}$

Categorical distribution



$$Pr(x_i|y_i = k) = Pr(x_i|y_i = k; W_1) = \mathcal{N}(\mu_k, \sigma_k^2)$$

$$Pr(y_i|y_{i-1}) = Pr(y_i|y_{i-1} = k; W_2) = Cat[\lambda_k]$$

Learning with HMM

- Supervised learning
  - Relatively simple. We first isolate the part of the model that we want to learn.
  - For example, we might learn the parameters $Pr(x_i|y_i = k; W_1) = \mathcal{N}(\mu_k, \sigma_k^2)$
    - from paired examples of xi and yi.
  - We can learn these parameters in isolation using the ML, MAP, or Bayesian methods.
  - The same apply for $Pr(y_i|y_{i-1} = k; W_2) = Cat[\lambda_k]$

Learning with HMM

- Unsupervised learning
  - More challenging
  - Beyond the scope of this presentation (dedicated to supervised learning)
  - Require notion such as :
    - Expectation Maximization method
    - Forward-Backward method
    - Baum-Welch algorithm

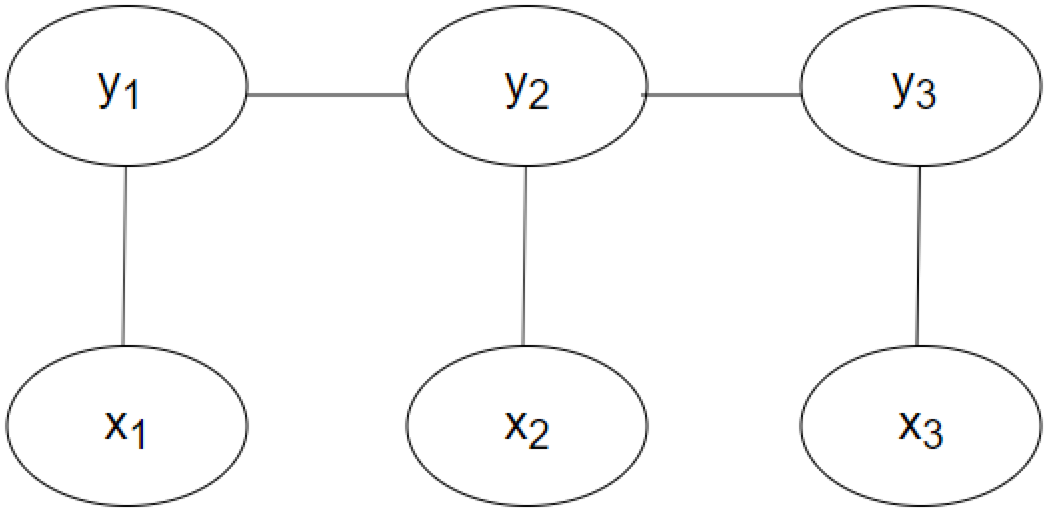
Limit of HMM

- Parameters (W_1et W_2) of the distributions are shared through time
  - All time steps have the same parameters
  - Corresponding to the assumption of a **stationary time series**.
- HMM are generative models
  - Discriminative models are more efficient to infer information on the world state (y)
- Although this is more general than the independence model, it is still very restrictive.

Directed and Undirected model

- HMM is a directed model
  - The tendency to observe the measurements x_i given that state y_i takes value k. → $\Pr(x_i|y_i = k)$
  - The current state is dependent on the previous one $\Pr(y_i|y_{i-1})$
- Undirected model
  - $\Pr(x_i|y_i = k)$ *is replaced by* $\phi(x_i, y_i)$: similarity function
    - Returns larger values when the measurements x_i and the world state are more compatible y_i.
    - Example : $\phi(x_i, y_i) = x_i . y_i$ *with* $x_i \in \{-1,1\}$ *and* $y_i \in \{-1,1\}$
  - $\Pr(y_i|y_{i-1})$ is replaced by $\zeta(y_i, y_{i-1})$: similarity function
    - Example : $\zeta(y_i, y_{i-1}) = y_i \, . y_{i-1_i}$ *and* $y_i \in \{-1,1\}$
    - Returns larger values when the adjacent states are more compatible.

1D Markov Random Field (MRF)



$$Pr(y_1, \cdots, y_M; x_1, \cdots, x_M) = \frac{1}{Z}\left[\prod_{i=1}^{M} \phi(x_i, y_i)\right]\left[\prod_{i=2}^{M} \zeta(y_i, y_{i-1})\right]$$

Z is a normalizing factors which form the partition function

Equivalence of models : HMM = 1D MRF

$$Z = \left[\prod_{i=1}^{M} z_i\right]\left[\prod_{i=2}^{M} z'_i\right]$$

$$z_i = \frac{\phi(x_i, y_i)}{Pr(x_i|y_i)}$$

$$z'_i = \frac{\zeta(y_i, y_{i-1})}{Pr(y_i|y_{i-1})}$$

Conditional Random Field (CRF) : A special case of MRF

$$Pr(y_1, \cdots, y_M | x_1, \cdots, x_M) = \frac{1}{Z}\left[\prod_{i=1}^{M} \phi(x_i, y_i)\right]\left[\prod_{i=2}^{M} \zeta(y_i, y_{i-1})\right]$$

$$Z = \sum_{y_1 \in \mathcal{Y}} \sum_{y_2 \in \mathcal{Y}} \sum_{y_3 \in \mathcal{Y}} \cdots \sum_{y_M \in \mathcal{Y}} \left[\prod_{i=1}^{M} \phi(x_i, y_i)\right]\left[\prod_{i=2}^{M} \zeta(y_i, y_{i-1})\right]$$

$$\mathcal{Y} = \{1, 2, 3, 4, 5\}$$

The x variables are given (observed/fixed).

Conclusion

- We have seen :
  - How to take into account dependencies from a data set
    - Sequential dependencies (time dependencies)
    - HMM model
      - A generative model
  - How to infer world states (labels y) from a given sequence x
    - Thanks to the maximum of the posterior distribution (Maximum A Posteriori : MAP)
  - How supervised learning could be achieved