



INSTITUT NATIONAL  
DES SCIENCES  
APPLIQUÉES  
RENNES

Eric Anquetil

INSA

Département Informatique

Version V2.0

[eric.anquetil@irisa.fr](mailto:eric.anquetil@irisa.fr)

[www.irisa.fr/intuidoc](http://www.irisa.fr/intuidoc)

# Analysis, Interpretation and Recognition of 2D (touch) and 3D (Gestures) for New Man-Machine Interactions



<b>CHAP. 1 :</b>	Introduction: understand the problematic of gesture interaction	8	— The overall pattern recognition process : Pattern Recognition	38
—	2D and 3D Action/Gesture recognition: a challenge ?	8	— The overall pattern recognition process: Pattern Recognition	39
—	2D gesture sensors: pen-based and touch-based gestures	9	— The overall pattern recognition process: Pattern Recognition	40
—	2D gesture sensors : pen-based and touch-based gestures	10	— The overall pattern recognition process (segmented gestures): Pattern Recognition	41
—	3D gesture sensors: whole body gestures recognition	11	<b>CHAP. 6 :</b>	Gesture classification: "Time-series" approaches
—	3D gesture sensors: Hand Gesture	12	—	Gesture classification: "Time-series" approaches
<b>CHAP. 2 :</b>	Inputs: time-series	14	—	Gesture classification: "Time-series" approaches
—	2D gesture inputs: Pen-based and touch-based gestures	15	<b>CHAP. 7 :</b>	Hidden Markov Model
—	2D gesture inputs: Multi-stroke and Multi-touch	16	—	Classification: Hidden Markov Models (HMM)
—	3D gesture inputs	17	—	HMM: Definition
—	Inputs: Trajectories / One generic approach for 2D/3D gesture recognition?	19	—	HMM: example
<b>CHAP. 3 :</b>	Introduction to Gesture Interaction	20	—	HMM: example
—	Gesture interaction	21	—	HMM: basic problems
—	Gesture interaction: Mono Stroke	22	—	HMM: Viterbi algorithm
—	Gesture interaction: Multi-Stroke	23	—	HMM: Viterbi algorithm
—	Gesture interaction	24	—	HMM: discrete versus continuous
—	Gesture interaction: Direct and Indirect commands	25	—	HMM: example
—	Gesture interaction: Direct and Indirect commands	26	—	HMM: for Speech
—	Gesture interaction: Direct and Indirect commands	27	<b>CHAP. 8 :</b>	Dynamic Time Warping (DTW)
—	Gesture interaction: multi-touch gestures	28	—	DTW : Introduction
—	Gesture interaction: Multi-user interaction	30	—	DTW : Principles
—	Perspective : future of Pen and Touch interaction [Pfeuffer CHI 2017]	31	—	DTW : Algorithm
<b>CHAP. 4 :</b>	Intra/inter -class variability (shape, spatial and temporal)	32	—	DTW : Illustration
—	Intra/Inter -class: shape variabilities	33	—	DTW : Pattern Recognition using KNN (without learning)
—	Intra/Inter -class: shape variabilities	35	—	DTW : Pattern Recognition using KNN (without learning)
—	Intra/Inter -class: temporal and spatial variabilities	36	—	DTW : Pattern Recognition using KNN (without learning)
—	Gesture Recognition: a transversal challenge	37	—	DTW : Pattern Recognition using KNN (without learning)
<b>CHAP. 5 :</b>	Gesture recognition: Isolated Gestures Classification (segmented)	38	—	DTW : Pattern Recognition using KNN (without learning)
—	2D and 3D Action/Gesture recognition: a challenge ?	39	—	DTW : Pattern Recognition using KNN (without learning)
—	Overview of the task: generic flowchart	40	—	DTW : Pattern Recognition with creating models (Learning phase)
—	The overall pattern recognition process (segmented gestures)	41	—	DTW : Pattern Recognition with creating models: Offline learning

—	DTW: Averaging of two signals to create a model	69	<b>CHAP. 10 :</b>	Non-segmented Action Recognition: Skeleton based and "Statistical" approaches
—	DTW: Averaging of more than two time-series	70	—	2D and 3D Action/Gesture recognition: a challenge ?
—	DTW: Barycenter Averaging Algorithm (DBA)	71	—	Gesture recognition in real-time streaming (non segmented): overview
—	DTW : learning category-specific deformations ...	72	—	Gesture recognition in real-time streaming (non segmented): overview
—	DTW : learning category-specific deformations ...	73	—	Gesture recognition in real-time streaming (non segmented): step 1
—	DTW : For Gesture Analysis	74	—	Gesture recognition in real-time streaming (non segmented): step 2
—	DTW : a parallel with Edit distance computation	75	—	Gesture recognition in real-time streaming (non segmented): step 3
<b>CHAP. 9 :</b>	Pre-segmented Action Recognition: Skeleton based and "Statistical" approaches	76	—	Gesture recognition in real-time streaming (non segmented): step 3
—	Segmented pattern representation	77	—	Gesture recognition in real-time streaming (non segmented): step 3
—	Skeleton based Action Recognition based on 3D gesture trajectories	78	<b>CHAP. 11 :</b>	Presentation of experimental results using Kinect and Leap Motion
—	Pre-segmented Action Recognition (Skeleton based)	79	—	Gesture recognition in real-time streaming (non segmented): MSRC-12 Dataset
—	Action representation by 3DMM: Kinect based patterns: whole body actions	80	—	Gesture recognition in real-time streaming (non segmented): MSRC-12 Dataset
—	Action representation by 3DMM - Segmented pattern recognition: synthesis	81	—	Evaluation measure
—	Action representation by 3DMM - Segmented pattern recognition: synthesis	82	—	Gesture recognition in real-time streaming / segmented: (DHG) dataset
—	Action representation by 3DMM: Step 1 - Pre-processing	83	—	Gesture recognition in real-time streaming / segmented: (DHG) dataset
—	Action representation by 3DMM: Step 2 - temporal hierarchy	84	—	Gesture recognition in real-time streaming / segmented: (DHG) dataset
—	Action representation by 3DMM: Step 3 - : dealing with the set of 2D trajectories	85	—	Gesture recognition in real-time streaming / NON-segmented: (LMDHG) dataset
—	Action representation by 3DMM: Step 3 - Direct 2D features extraction	86	—	Gesture recognition in real-time streaming / NON-segmented: (LMDHG) dataset
—	Action representation by 3DMM: Step 4 - statistical Learning and classification	87	—	Gesture recognition in real-time streaming / Segmented: (LMDHG) dataset
—	Action representation by 3DMM: Step 4 - statistical Learning and classification	88	—	Gesture recognition in real-time streaming / NON-segmented: (LMDHG) dataset
—	Action representation by 3DMM: Step 4 - statistical Learning and classification	89	<b>CHAP. 12 :</b>	Early Recognition
—	Action representation by 3DMM: Step 4 - statistical Learning and classification	90	—	2D and 3D Action/Gesture recognition: a challenge ?
—	Action representation by 3DMM: Some results on the HDM05 dataset	91	—	Gesture Early Recognition: Introduction
—	Evaluation / Validation: Cross-Validation	92	—	Gesture Early Recognition: Difficulties for Recognition
—	Some results of 3DMM approach on the HDM05 dataset	93	—	Gesture Early Recognition: A multi-classifier early recognition system
—	Pre-segmented Action Recognition (Skeleton based)	94	—	Gesture Early Recognition: Reject option
—	Action representation by HIF 3D: 3D features inspired by 2D features	95	—	Gesture Early Recognition: Reject option
—	Action representation by HIF 3D: 3D features inspired by 2D features	96	—	Gesture Early Recognition: Ambiguity rejection
—	Action representation by HIF 3D: 3D features inspired by 2D features	97	—	Gesture Early Recognition: Outlier rejection
—	Action representation by HIF 3D: 3D features inspired by 2D features	98	—	Gesture Early Recognition: consistance checking
—	Some results of HIF3D approach on the HDM05 dataset	99		

— Gesture Early Recognition: Experiments	131	— Neural Networks	162
— Gesture Early Recognition: Experiments	132	— MultiLayer Perceptron (MLP)	163
<b>CHAP. 13 : Fuzzy Clustering</b>		— MLP: Learning and generalization	164
— Fuzzy clustering: Introduction	134	— MLP: Universal approximator	165
— Fuzzy clustering: Examples	135	— MLP: Learning	166
— Fuzzy clustering: Outline	136	— MLP: Back-propagation (BP) algorithm	167
— Fuzzy clustering: Data Representation	137	— MLP: Back-propagation (BP) algorithm	168
— Fuzzy clustering: Data Representation	138	— MLP: Back-propagation (BP) algorithm	169
— Fuzzy clustering: Different clustering families	139	— MLP: Back-propagation (BP) algorithm	170
— Fuzzy clustering: Alternating Clustering methods	140	— MLP: Learning with validation	171
— Fuzzy clustering: Hard C-Means	141	— MLP: knowledge modeling	172
— Fuzzy clustering: Constrained crisp partition	142	— Radial-Basis Function Neural Networks (RBFNN)	173
— Fuzzy clustering: Hard C-Means	143	— RBFNN : Learning	174
— Fuzzy clustering: Fuzzy C-Means	144	<b>CHAP. 17 : Reject Option</b>	
— Fuzzy clustering: Fuzzy partition	145	— Reject option	176
— Fuzzy clustering: Fuzzy C-Means	146	— Reject option with thresholds	177
— Fuzzy clustering: Fuzzy C-Means	147	— Distance reject / with thresholds	178
— Fuzzy clustering: Fuzzy C-Means / example	148	— Confusion reject / with thresholds	179
— Fuzzy clustering: Possibilistic clustering	149	— Reject option: Main approaches (distance reject)	180
— Fuzzy clustering: Possibilistic Clustering	150	— Evaluation of Recognition Systems	181
— Fuzzy clustering: Cluster validity	151	— Evaluation: distance reject / evaluation	182
— Fuzzy clustering: two different goals	152	— Evaluation: confusion reject / evaluation	183
— Fuzzy clustering: Shell clustering	153	<b>CHAP. 18 : Support Vector Machines</b>	
— Fuzzy clustering: Examples	154	— Basic notion of Support-Vector-Machines (SVM)	185
<b>CHAP. 14 : Classification: Linear Discriminant Functions</b>		— Basic notion of Support-Vector-Machines (SVM)	186
— Classification with Linear Discriminant Functions	156	— Basic notion of Support-Vector-Machines (SVM)	187
— Classification with Linear Discriminant Functions	157	— Basic notion of Support-Vector-Machines (SVM)	188
— Linear Discriminant Functions – Generalization	158	— Basic notion of Support-Vector-Machines (SVM)	189
— Linear Discriminant Functions – Generalization	159		
— Linear Discriminant Functions – Generalization	160		
<b>CHAP. 16 : Neural Networks</b>			

## \_Chapitre 1

### Introduction: understand the problematic of gesture interaction

#### \_Chap. 1 | 2D and 3D Action/Gesture recognition: a challenge ?

8

- Introduction: understand the problematic of gesture interaction
  - What is a gesture: the different natures of gestures
  - Human Computer Interaction: new opportunities
- Gesture recognition: Isolated Gestures Classification (segmented)
  - Overview of the task: recognizing isolated gestures (The overall pattern recognition process)
  - Machine Learning and Pattern recognition: a short overview of some existing techniques
    - Gesture classification: "Time-series" approaches
    - Pre-segmented Action Recognition: Skeleton based and "Statistical" approaches
- Gesture recognition in real-time streaming (non segmented)
  - Overview of the task: recognizing in real-time streaming
  - Non-segmented Action Recognition: Example of one approach [Boulahia 2017]
  - Presentation of experimental results using Kinect and Leap Motion
- Early Gesture recognition

## ■ Pen-based gesture interaction

### ■ Device platforms

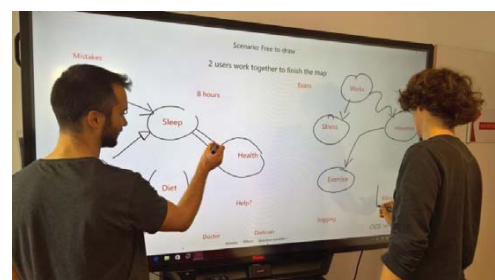
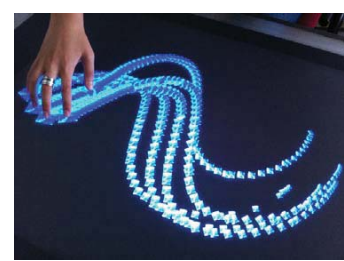
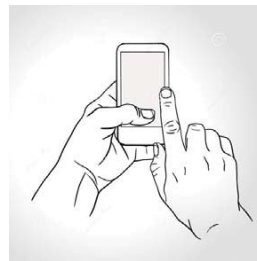
- Smartphone
- Digital Pen
- Tablet PC
- Electronic Whiteboard
- ...



## ■ Touch-based gesture interaction (touch screen)

- Multi touch based interaction (ex: whiteboarding solution...)
- Multi-user based interaction (ex: surface table, surface Hub...)

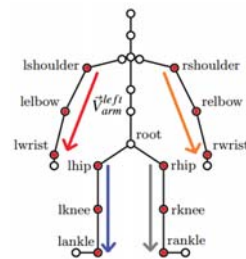
## ■ Tracking technology: capacitive touch screen display, ultrasound, infrared...



- Dynamic whole body gestures recognition
  - Wide range of application fields: such as video surveillance, sport video analysis, human-computer interaction, computer animation and even health-care.

- Two main groups of approaches
  - RGB + Depth image recognition
  - Skeleton-based action recognition

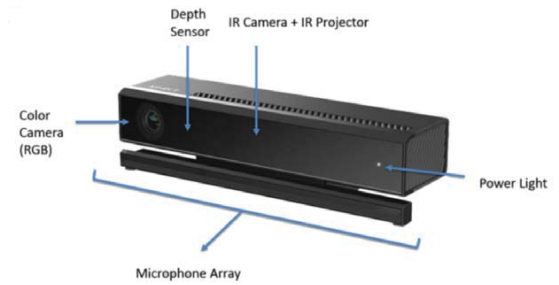
- Sensor technologies
  - Emergence of Kinect like sensors (2010)



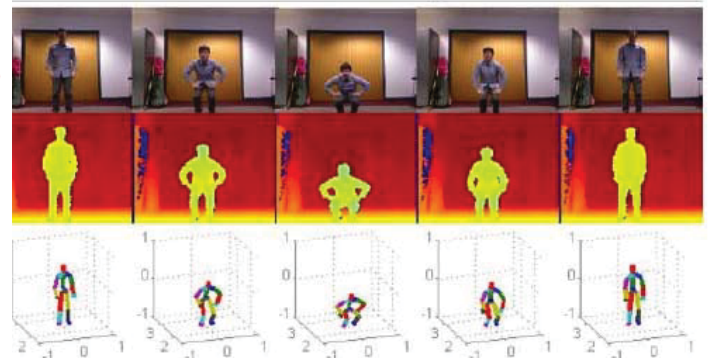
RGB

Depth

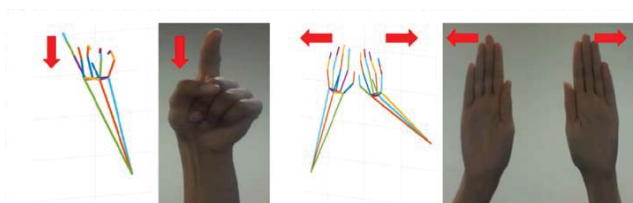
Skeleton



### Crouching



- Dynamic hand gestures
  - using skeleton joint data
- Sensor technologies
  - the Leap Motion device
  - Intel's RealSense depth-sensing 3D camera
  - Depth sensor + camera
- Few existing applications





## \_Chapitre 2

### Inputs: time-series

13

#### \_Chap. 2 | 2D gesture inputs: Pen-based and touch-based gestures

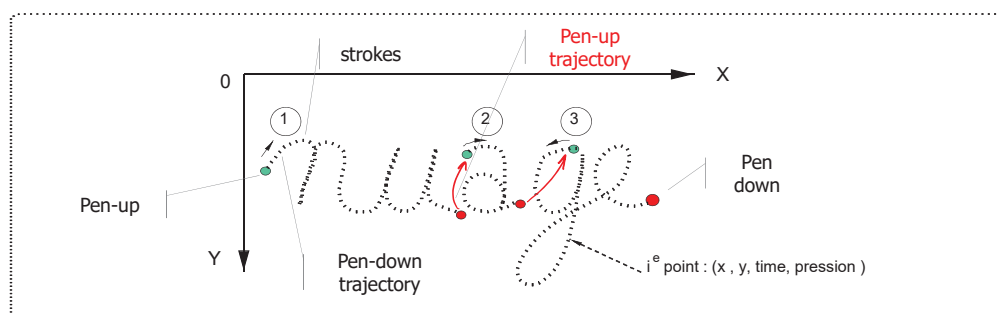
14

##### ■ On-line



##### ■ Data input

(x, y, time, pressure) / signal : sequences of 2D points

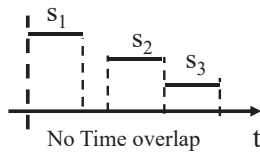


## \_Chap. 2 | 2D gesture inputs: Multi-stroke and Multi-touch

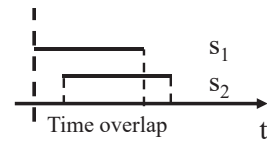
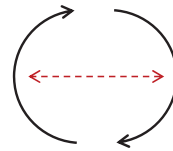
### ■ Multi-stroke and Multi-touch Gesture



**Multi-stroke  
(sequence of strokes)**



**Multi-touch  
(several strokes in //)**



### ■ Several trajectories to consider

#### ❖ Strokes are written in sequence

- Shape
- Spatial relation

#### ❖ Strokes are synchronized or partial synchronized

- Shape
- Spatial relation
- Temporal relation

## \_Chap. 2 | 3D gesture inputs

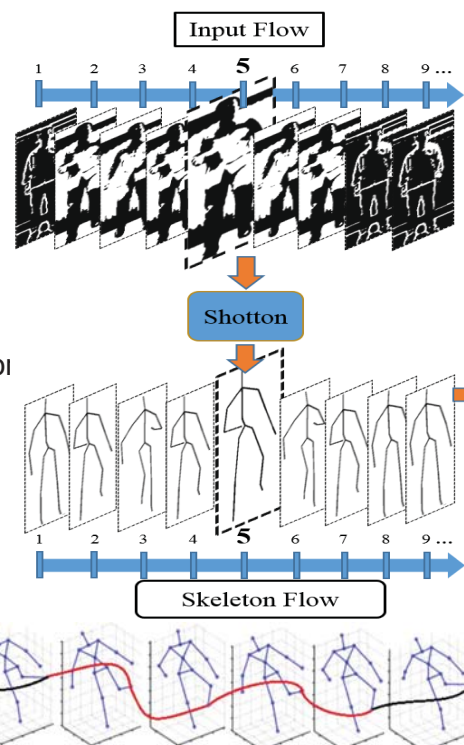
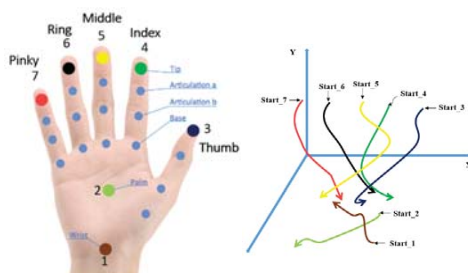
### ■ Two main groups of approaches:

#### ■ RGB-D based => input data = a sequence of frames



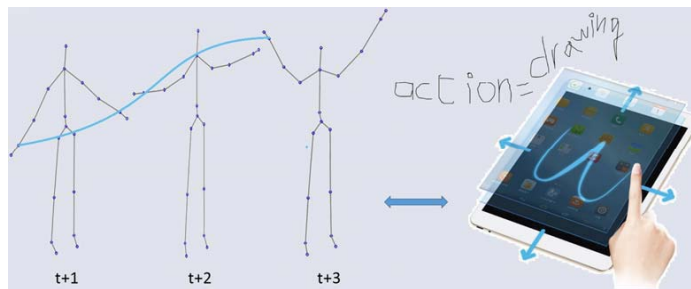
#### ■ Skeleton based

- By using Kinect, LeapMotion
- a sequence of 3D points = trajectory, angular information





- 3D gesture
  - A robust approach : Skeleton based approach
    - capture the essential structure of a subject in an easily understandable way
    - robust to variations in viewpoint and illumination
  - skeleton data consist in trajectories of the body joints
- Trajectories: a unified way to consider gestures
  - Same data type: trajectories or signal
  - 3D gesture trajectories may be processed similarly to 2D trajectories
- Moreover from Graphonomic point of view
  - 3D and 2D gestures : a human is the performer

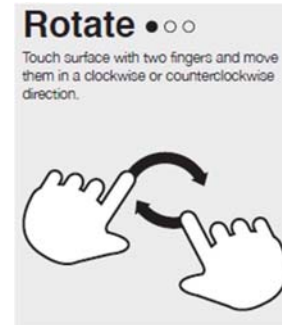
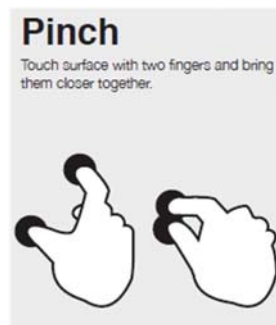


© eric.anquetil@irisa.fr

## \_Chapitre 3

### Introduction to Gesture Interaction

- General Introduction based on [Zhaoxin Chen 2016]
  - Touch gesture examples[1]

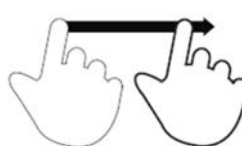


[1] Touch gesture reference guide, Luke Wroblewski, <http://www.lukew.com/>

- Development of gesture interaction



Tap

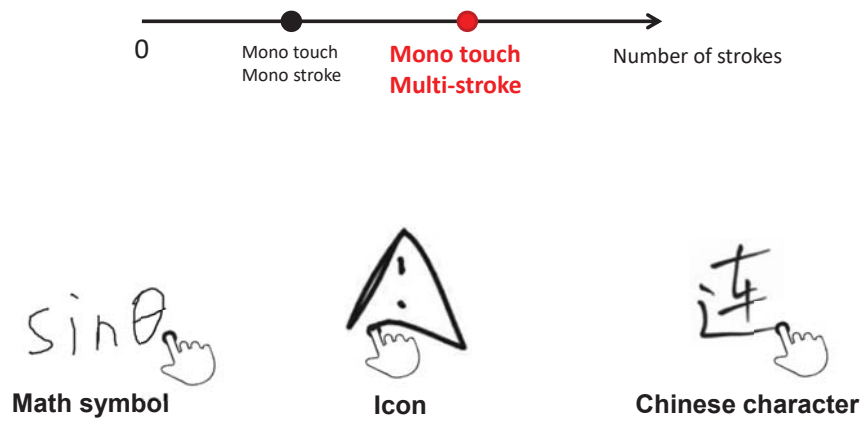


Drag

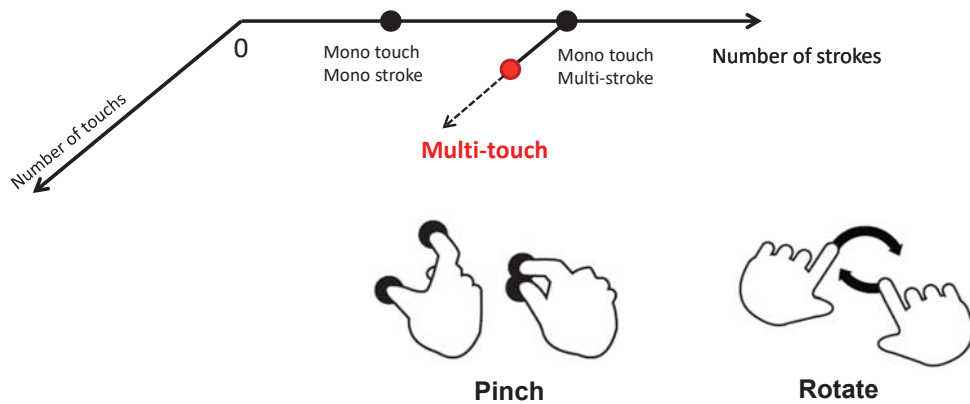


Handwritten character

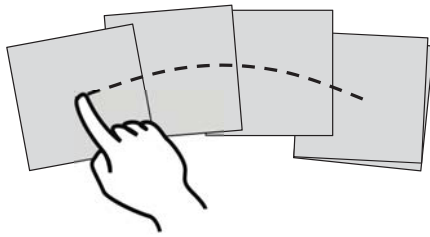
■ Development of gesture interaction



■ Development of gesture interaction



- Two types of interactions

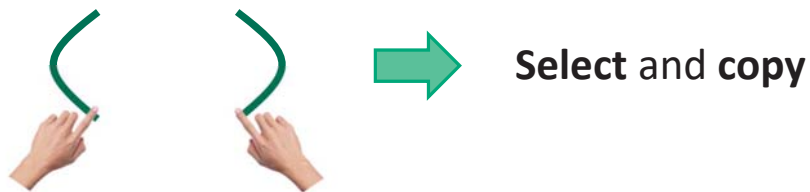


Direct manipulation



Indirect command

- What if a user wants to use the multi-touch gesture to make a command instead of manipulation.



**Select and copy**



**Paste at somewhere**

How to recognize a multi-touch gesture as indirect command?

- Is it possible to merge these two interactions into a same interface



**Pinch**

Direct manipulation

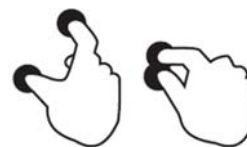
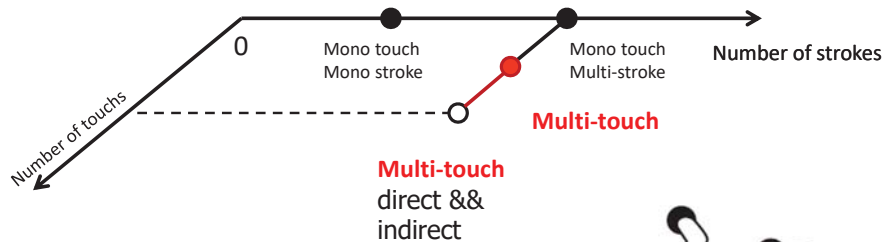


**Paste**

Indirect command

How to support these two interactions in a same context?

- Open more possibilities to use multi-touch gestures
  - complex gesture for indirect commands
  - mix the direct manipulation and indirect command



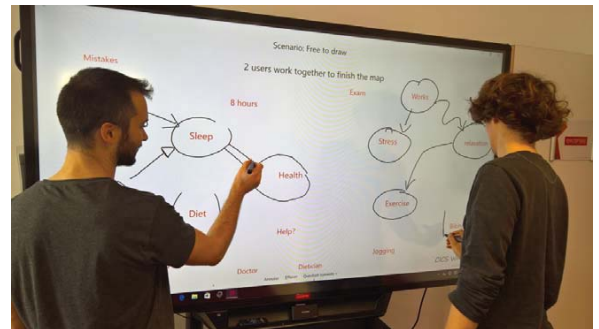
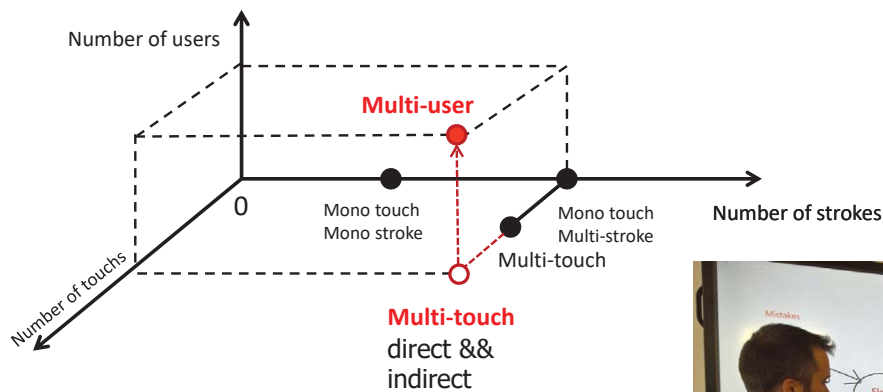
**Pinch**



**Paste**

### ■ Multi-user interaction

- to deal with several gestures in the same time



© eric.anquetil@irisa.fr

### ■ Example of novel way of interaction: Thumb + Pen interactions

- Support simultaneous pen and touch interaction, with both hands
- allow changing the mode of the pen
- changing the mode that applies to the pen conventions.
- additional navigation functionality
- ...



**Figure 1: Thumb + Pen interaction enables simultaneous bimanual pen+touch while holding a tablet with the off-hand.**

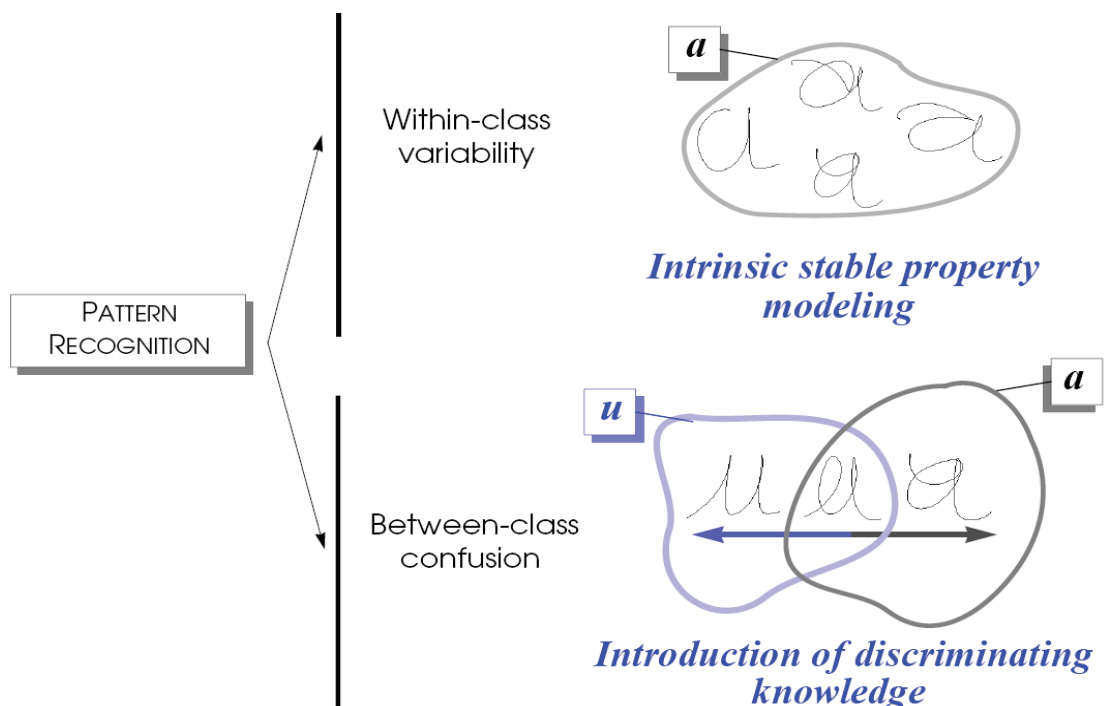
[Pfeuffer 2017] *Thumb + Pen Interaction on Tablets*  
Ken Pfeuffer, Ken Hinckley, Michel Pahud, Bill Buxton  
Microsoft Research, Redmond, WA, USA  
Interactive Systems, Lancaster University, UK

© eric.anquetil@irisa.fr



## \_Chapitre 4

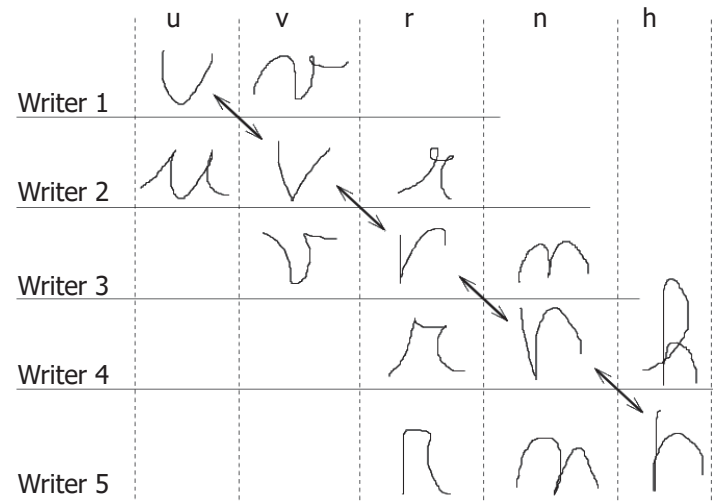
### Intra/inter -class variability (shape, spatial and temporal)



■ Writer dependent versus Writer-independent recognizer

- Resource cost
- Ambiguity of characters between different writers
- No ambiguity for each writer

■ [Mouchère07]



\_Chap. 4 | Intra/Inter –class: temporal and spatial variabilities

■ Temporal variability

- Occurs when subjects perform gestures with different speeds

■ Inter-class spatial variability

- Different gesture classes are likely to result in different amount of displacements

■ Intra-class spatial variability

- Same action class with different amount of displacements
- In some applications, capturing such intra-class variabilities might be desirable as it brings additional information and could allow for different interpretations of the same class of gesture. Otherwise need to must be neutralized.

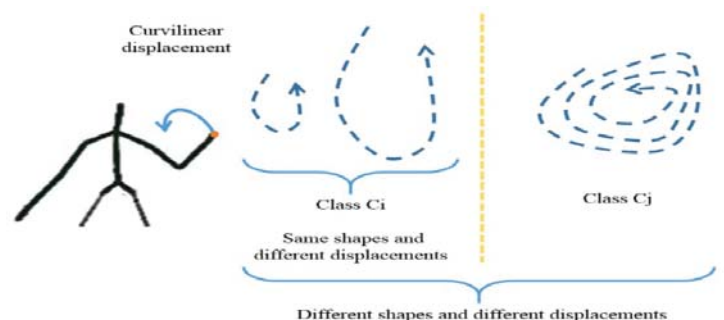
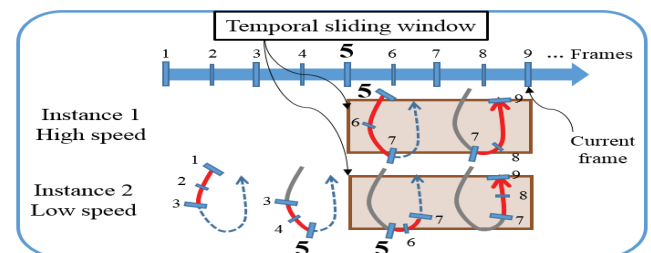
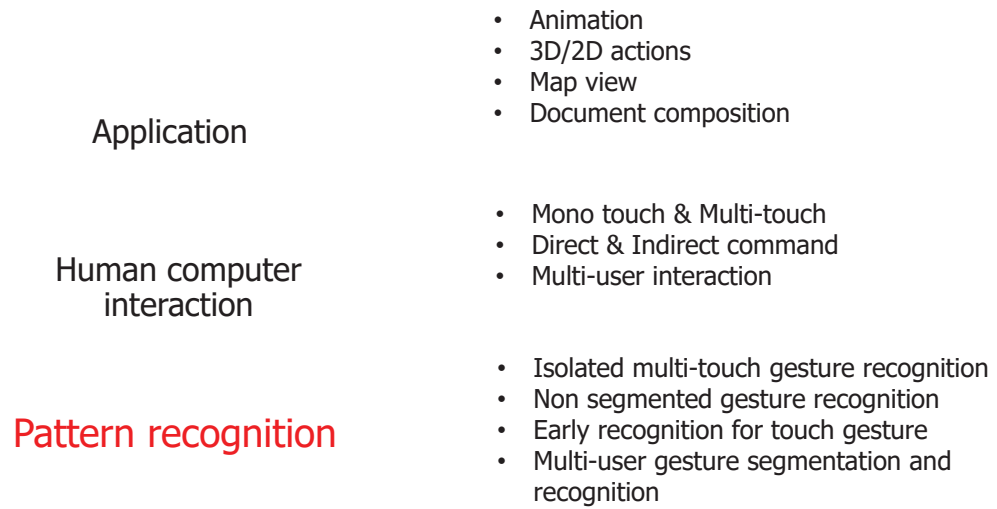


Fig. 1. Illustration with a single joint trajectory of intra-class spatial variability within a class  $C_i$  (left) and inter-class spatial variability between  $C_i$  and  $C_j$  (right).



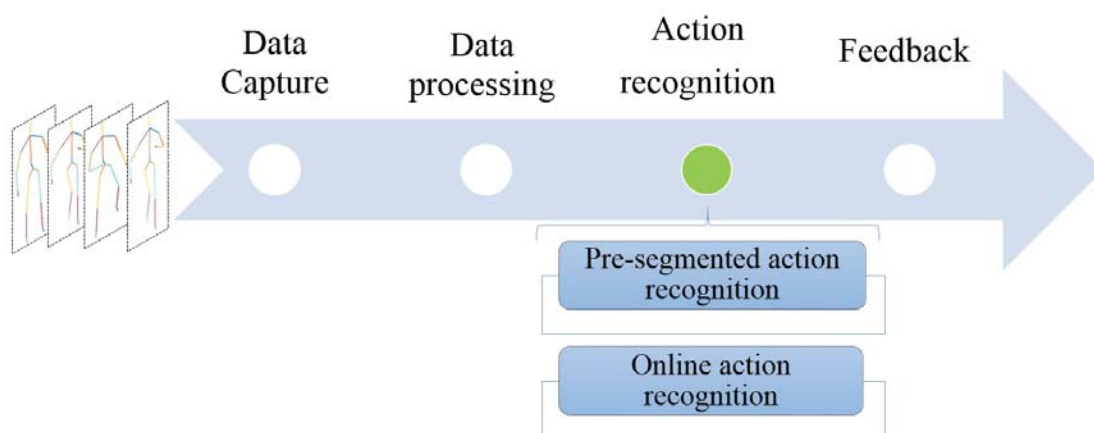
## \_Chapitre 5

### Gesture recognition: Isolated Gestures Classification (segmented)

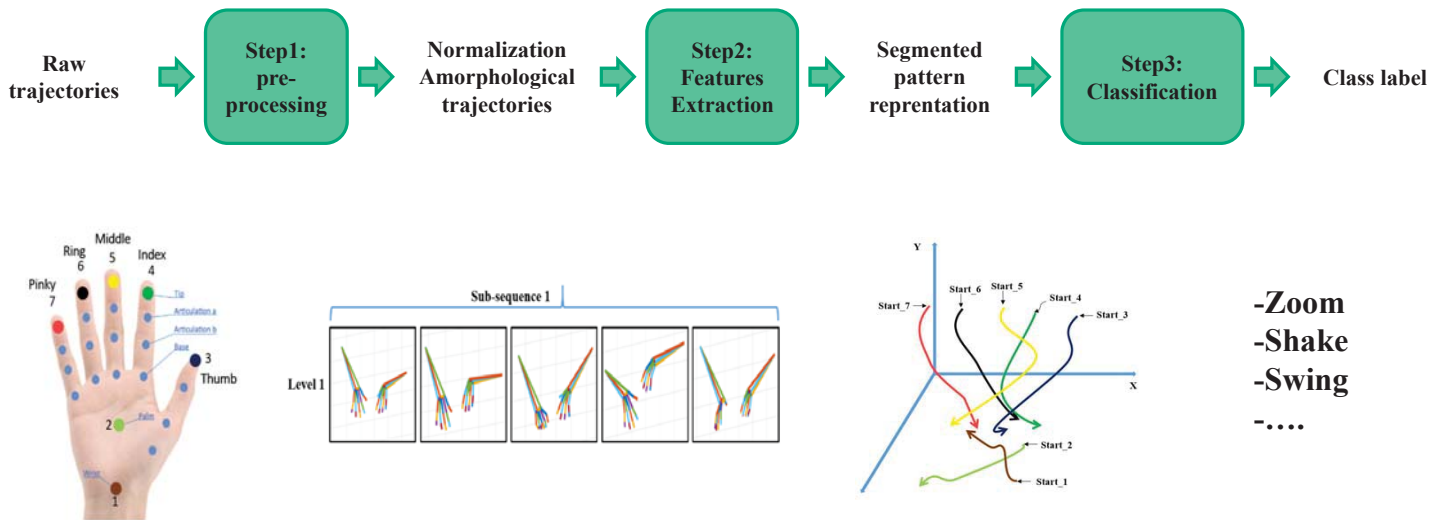
- Introduction: understand the problematic of gesture interaction
  - What is a gesture: the different natures of gestures
  - Human Computer Interaction: new opportunities
- Gesture recognition: Isolated Gestures Classification (segmented)
  - Overview of the task: recognizing isolated gestures (The overall pattern recognition process)
  - Machine Learning and Pattern recognition: a short overview of some existing techniques
    - Gesture classification: "Time-series" approaches
    - Pre-segmented Action Recognition: Skeleton based and "Statistical" approaches
- Gesture recognition in real-time streaming (non segmented)
  - Overview of the task: recognizing in real-time streaming
  - Non-segmented Action Recognition: Example of one approach [Boulahia 2017]
  - Presentation of experimental results using Kinect and Leap Motion
- Early Gesture recognition

\_Chap. 5 | Overview of the task: generic flowchart

❖ **Human action recognition**



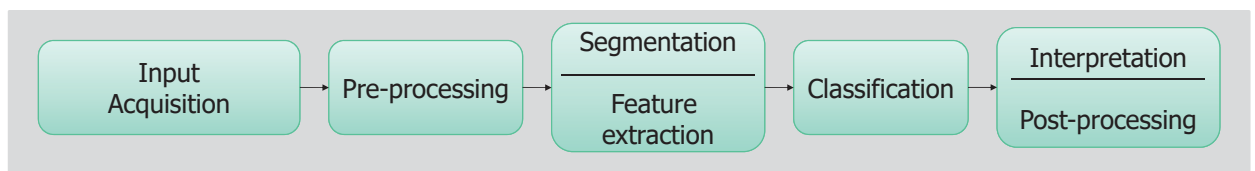
- The overall process for segmented dynamic gesture recognition (hand gesture illustration)



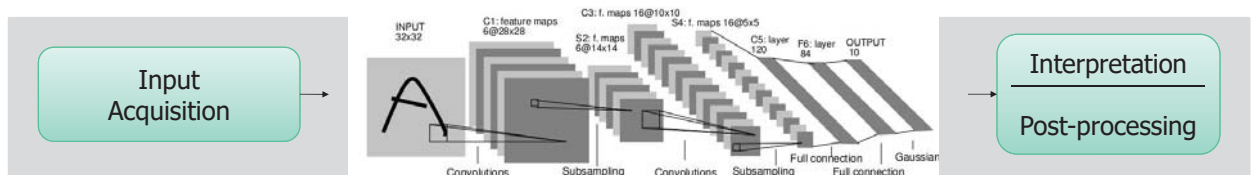
- Overview



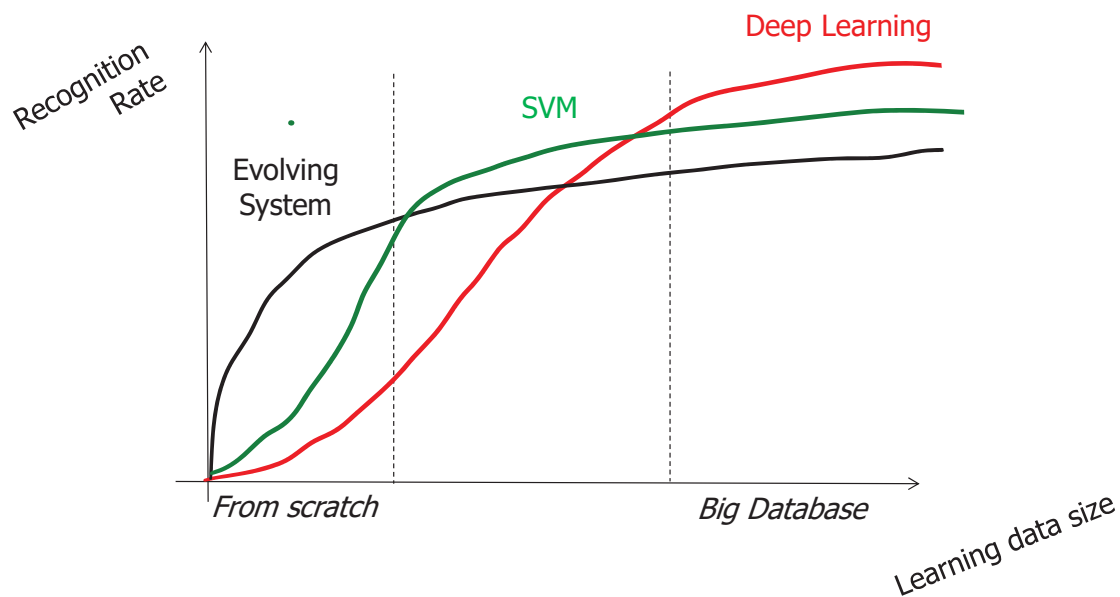
- Classical Process



- With Deep Learning



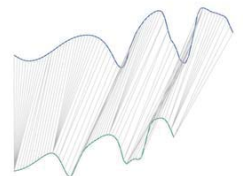
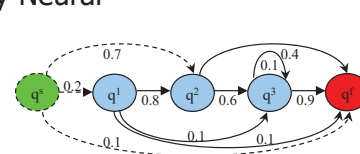
## Overview



## Chap. 5 | The overall pattern recognition process: Pattern Recognition

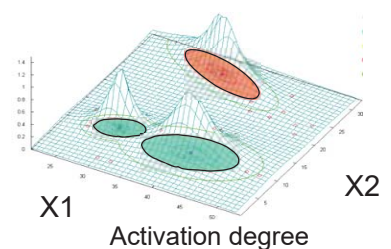
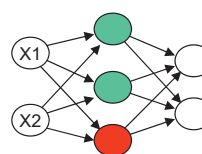
### “Dynamic /Time-series” approaches (AMRG-AIR)

- Input : Handle the sequential data with variable lengths
  - Elastic Matching (Dynamic Time Wrapping, DTW) → similarity between two sequences
  - Hidden Markov Model (HMM)
  - Recurrent neural networks (RNNs), Time, Space Delay Neural Network (TDNN, SDNN)
  - long short-term memory (LSTM) network



### “Static” approaches (ATI)

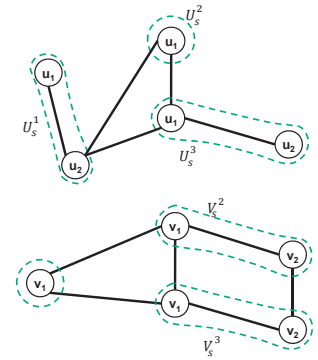
- Input : Feature vector (low level representation)
  - Recognition system: Classifier (learning and generalization phase)
    - Support Vector Machine (SVM)
    - Neural Network (MLP, RBF,...),
    - Fuzzy Inference System (FIS),
    - Decision tree, ...
  - CNN: Convolutional Neural Networks





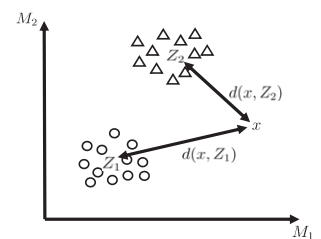
## ■ "Structural" approaches

- Input
  - Primitives  $\rightarrow$  feature vector (high level representation)
  - Based on fine analysis of the pattern
- Recognition system: Classifier (learning and generalization phase)
  - Possibly the same classifier as "statistical" approaches
  - Fuzzy Inference System (FIS), Decision Tree, ...
- Advantage: transparent system, possible optimization  
Drawback : more difficult to design



## ■ Others

- K nearest neighbors (KNN) (without Learning phase ...)... need to define a distance (ex: DTW...)
- Hybrid Approaches : HMM + NN



© eric.anquetil@irisa.fr

# \_Chapitre 6

## Gesture classification: "Time-series" approaches

## ■ Many fields to consider time-ordered Series of Data:

### ■ Motion/Gesture

- M. Morel, C. Achard, R. Kulpa, and S. Dubuisson, "Automatic evaluation of **sports motion**: a generic computation of spatial and temporal errors", Image and Vision Computing, vol. 64, pp. 67–78, 2017.
- M. T. Pham, R. Moreau, and P. Boulanger, "Three-dimensional **gesture comparison** using curvature analysis of position and orientation," in EMBC'10, pp. 6345–6348, IEEE, 2010.
- F. Zhou and F. D. la Torre Frade, "Canonical time warping for alignment of **human behavior**," in Advances in Neural Information Processing Systems Conference (NIPS), December 2009.

### ■ Handwriting

- I. Guler and M. Meghdadi, "A different approach to off-line **handwritten signature** verification using the optimal dynamic time warping algorithm," Digital Signal Processing, vol. 18, no. 6, pp. 940–950, 2008.
- Mitoma, H., S. Uchida, and H. Sakoe. **Online character** recognition based on elastic matching and quadratic discrimination. in Eighth International Conference on Document Analysis and Recognition. 2005. p. 36-40 Vol. 31.
- Niels, R. and L. Vuurpijl, Dynamic time warping applied to **Tamil character** recognition. Eighth International Conference on Document Analysis and Recognition, 2005: p. 730-734 Vol. 732.

### ■ Biological systems

- B. S. Raghavendra, D. Bera, A. S. Bopardikar, and R. Narayanan, "**Cardiac** arrhythmia detection using dynamic time warping of ECG beats in e-healthcare systems," in IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, pp. 1–6, IEEE, 2011.

### ■ Audio (speech or music) signals.

- G. Kang and S. Guo, "Variable sliding window DTW **speech** identification algorithm," in Ninth International Conference on Hybrid Intelligent Systems, pp. 304–307, IEEE, 2009.
- Ning Hu, R. Dannenberg, and G. Tzanetakis, "Polyphonic audio matching and alignment for **music** retrieval," in IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 185–188, IEEE, 2003.

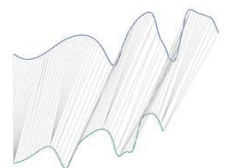
## ■ Time-series challenges

### ■ Difficulties: length variability

- requiring their temporal alignment as a pre-processing step

### ■ To learn a Model

- to derive a single model from a set of signals corresponding to several instances of the same physical process.

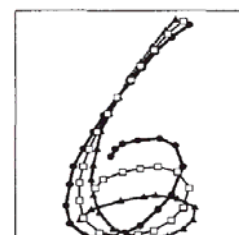


## ■ Main Simple Approaches

### ■ Hidden Markov Model (HMM)

### ■ Dynamic programming (DP) / Dynamic time warping (DTW)

- [Morel 2017] Marion Morel, Catherine Achard, Richard Kulpa, and Séverine Dubuisson. *Time-series averaging using constrained dynamic time warping with tolerance*. *Pattern Recognition*, 2017.



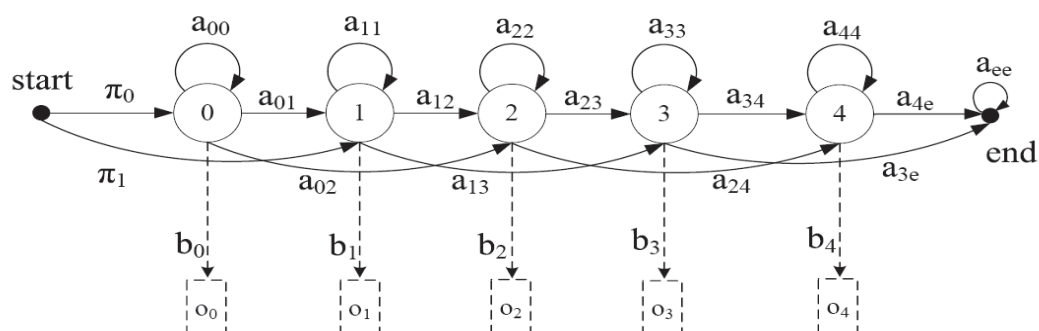
# \_Chapitre 7

## Hidden Markov Model

### \_Chap. 7 | Classification: Hidden Markov Models (HMM)

46

- Hidden Markov Models: approach inspired from speech recognition
  - deal with sequence of observations
  - find application in practically all ranges of the statistic pattern recognition
- HMMs
  - Generalization of homogeneous Markov chains with a stochastic process on **two stochastic processes**
    - Sequence of the states is produced by the transition probabilities  $a_{ij}$
    - At each state is associated an emission probability  $b_j(o)$



## Definition

- An HMM is a double stochastic process
  - an underlying stochastic process generates a sequence of states

$$q_1, q_2, \dots, q_p \dots q_T$$

Where  $t$  : discrete time, regularly spaced  $T$  : length of the sequence  
 $q_t \in Q = \{q_1, q_2, \dots, q_N\}$   $N$  : the number of states

- each state emits an observation according to a second stochastic process :
  - $o_t \in O = \{o_1, o_2, \dots, o_M\}$   $M$  : number of symbols
  - $o_t$  : a discrete symbol

## Specification of an HMM $\lambda = (\Pi, A, B)$

- $A$  - the state transition probability matrix

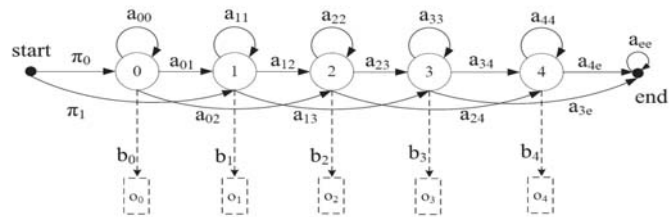
$$a_{ij} = P(q_{t+1} = j | q_t = i)$$

- $B$  - observation probability distribution

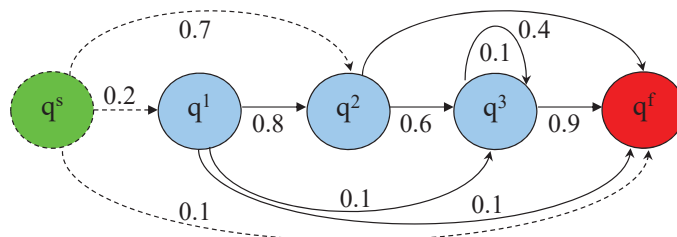
$$b_{ij} = P(o_t = o^j | q_t = q^i)$$

- $\Pi$  - the initial state distribution

$$b_{ij} \geq 0 \quad \text{and} \quad \sum_{j=1}^M b_{ij} = 1$$



## Example of non ergodic model (left-right model)



- 3 states + 1 starting state  $q^s$  + 1 final state  $q^f$ 
  - $q^s$  and  $q^f$  are non emitting states
- Assume there are 2 symbols to observe  $O = \{o^1=a, o^2=b\}$ 
  - Example of possible observation sequence: "a b b b"

$$\Pi = \begin{bmatrix} 0.2 \\ 0.7 \\ 0 \\ 0.1 \end{bmatrix}$$

Initiale state probabilities

$$A = \begin{bmatrix} 0 & 0.8 & 0.1 & 0.1 \\ 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0.1 & 0.9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Transition state probabilities

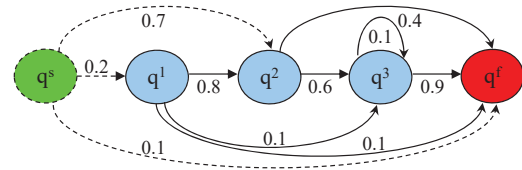
$$B = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \\ 0.1 & 0.9 \end{bmatrix} \quad \begin{matrix} P(a|q^1) \\ P(b|q^3) \end{matrix}$$

Observation symbol probabilities

[C. Viard-Gaudin]

■ The most probable state sequence is:

- $q^2, q^3$  resulting in the symbol sequence “bb”.
- But this sequence can also be generated by other state sequences, such as  $q^1, q^2$ .



$$B = \begin{bmatrix} a & b \\ 0.8 & 0.2 \\ 0.4 & 0.6 \\ 0.1 & 0.9 \end{bmatrix}$$

■ Computation of the likelihood of an observation sequence:

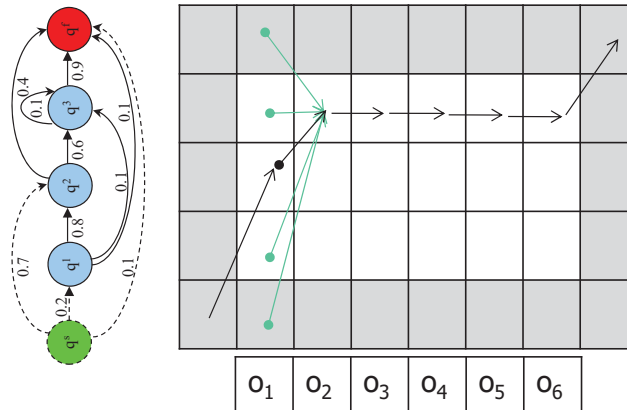
- Given  $X = \text{“aaa”}$  compute the likelihood for this model :  $P(\text{aaa} | \lambda)$
- The likelihood  $P(X | \lambda)$  is given by the sum over all possible ways to generate  $X$ .

State sequence	Init	Obs a	Trans	Obs a	Trans	Obs a	Trans	Joint probability
$q^1 q^2 q^3$	0.2	0.8	0.8	0.4	0.6	0.1	0.9	0.0027648
$q^1 q^3 q^3$	0.2	0.8	0.1	0.1	0.1	0.1	0.9	0.0000144
$q^2 q^3 q^3$	0.7	0.4	0.6	0.1	0.1	0.1	0.9	0.0001512
$P(\text{aaa}   \lambda) =$								0.0029304

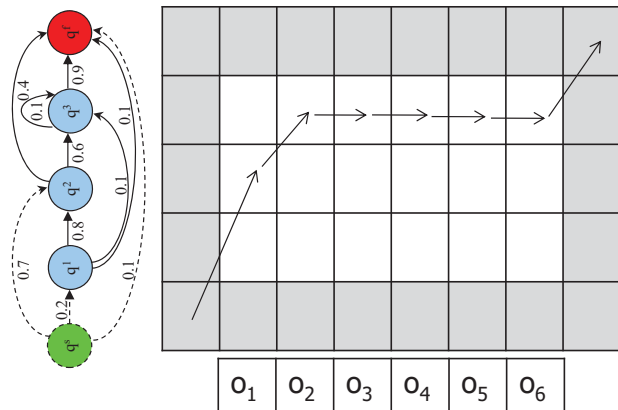
■ The 3 basic problems for HMMs

- Problem 1 : **Evaluate** the probability of an observation sequence (Forward-Backward algorithm)
  - Given  $O = (o_1, o_2, \dots, o_T)$  and a model  $\lambda$
  - How to efficiently compute the probability  $P(O | \lambda)$  of a given observation sequence?
- Problem 2 : **Find out the most likely state sequence** (Viterbi algorithm)
  - Given  $O = (o_1, o_2, \dots, o_T)$  and a model  $\lambda$
  - how to efficiently find the optimal state sequence for which the probability of a given observation  $O = (o_1, o_2, \dots, o_T)$  is maximum.
- Problem 3 : **Learning** (Baum-Welch algorithm)
  - Given a set of training sequences  $\{O = (o_1, o_2, \dots, o_T)\}$ , how to efficiently estimate the parameters of a model  $\lambda = (\Pi, A, B)$  according to the maximum likelihood criterion.

- Viterbi algorithm: Solution by Dynamic Programming
  - Define  $\delta_t(i)$  the highest probability path ending in state  $q^i$ 
    - $\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P(q_1, q_2, \dots, q_t = q^i, o_1, o_2, \dots, o_t | \lambda)$
  - By induction:
    - $\delta_{t+1}(k) = \max_{1 \leq i \leq N} [\delta_t(i) a_{ik}] \cdot b_k(o_{t+1})$ , with  $1 \leq k \leq N$
    - Memorize  $\Psi_{t+1}(k) = \arg \max_{1 \leq i \leq N} (\delta_t(i) a_{ik})$



- Viterbi algorithm: Solution by Dynamic Programming
  1. Initialization
    - For  $1 \leq i \leq N$   $\{ \delta_1(i) = \pi_i \times b_i(o_1); \Psi_1(i) = 0; \}$
  2. Recursive computation
    - For  $2 \leq t \leq T$ 
      - For  $1 \leq j \leq N$ 
        - $\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \cdot b_j(o_t);$
        - $\Psi_t(j) = \arg \max_{1 \leq i \leq N} (\delta_{t-1}(i) a_{ij});$
  3. Termination
    - $P^* = \max_{1 \leq i \leq N} [\delta_T(i)];$
    - $q^*_{T-1} = \arg \max_{1 \leq i \leq N} [\delta_T(i)];$
  4. Backtracking
    - For  $t=T-1$  down to 1  $\{ q^*_t = \Psi_t(q^*_{t+1}); \}$



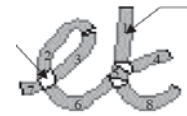
$P^*$  gives the required state-optimized probability  
 $\Gamma^* = (q_1^*, q_2^*, \dots, q_T^*)$  is the optimal state sequence



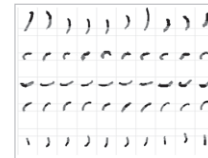
■ Different types of HMMs on the basis of the kind of symbols:

■ Discrete HMMs

- Number of possible symbols, probability of the symbols in matrix
- quantization errors at boundaries
- relies on how well Vector Quantization (clustering) partitions the space
- sometimes problems estimating probabilities when unusual input vector not seen in training



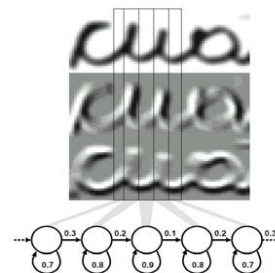
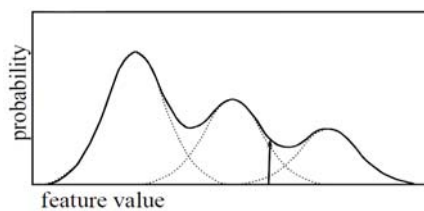
Sequence of primitives  
[Viard Gaudin]



Discrete HMM  
5 clusters  
[Viard Gaudin]

■ Continuous HMMs

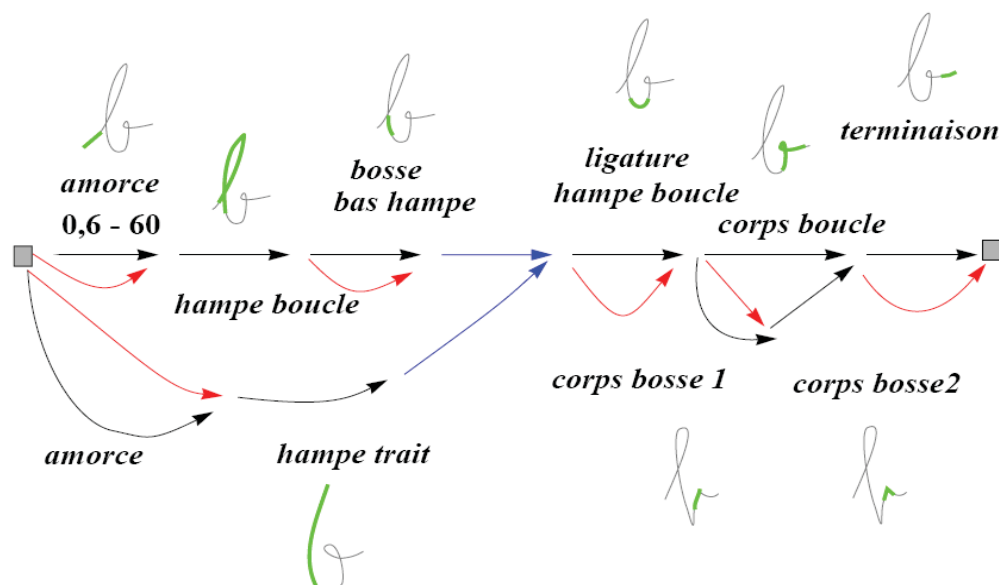
- Probabilities of symbols in continuous form; distribution density
  - Example: the emission probability is expressed with mixtures of Gaussians.



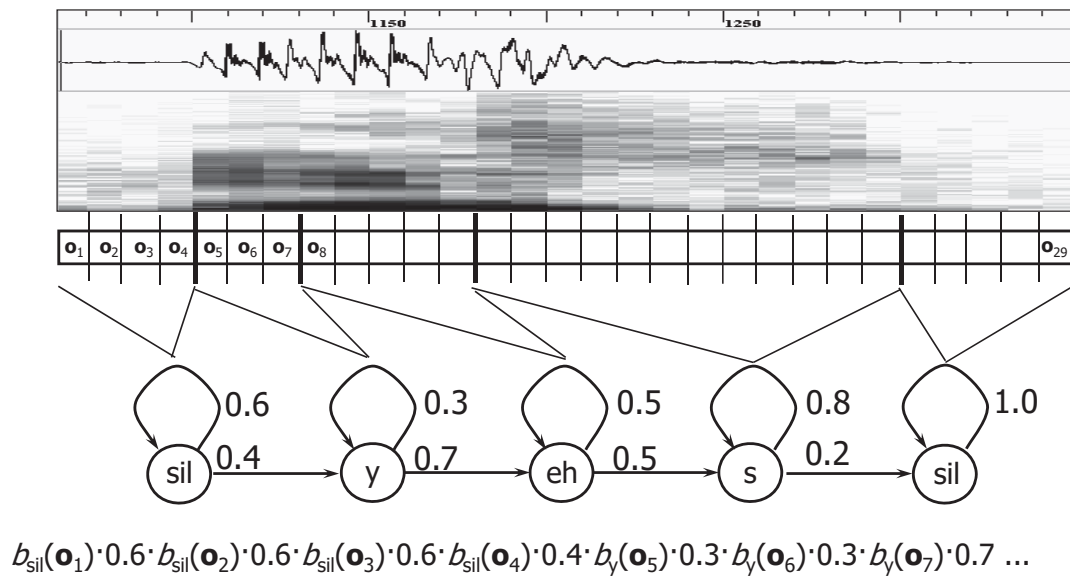
[Juan 04]

■ Another explicit segmentation : example of an on-line approaches

- Discrete Emission probability
  - Sequence based on primitives

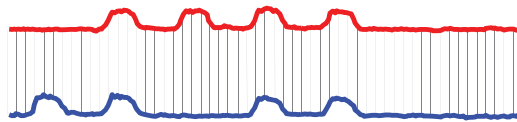


- Example of using HMM for word "yes" [John-Paul Hosom 2009]

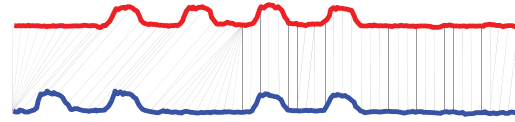


## \_Chapitre 8

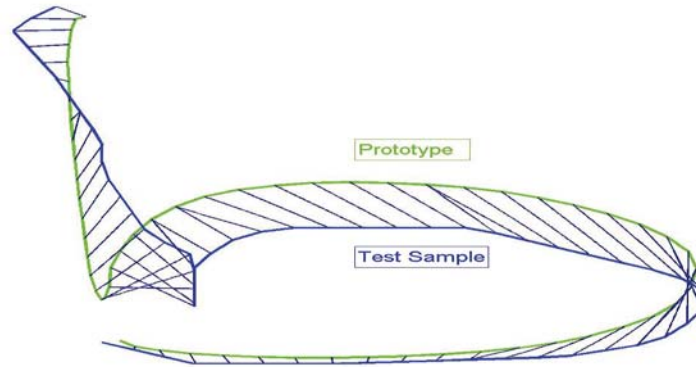
### Dynamic Time Warping (DTW)



**Euclidean Distance**  
Sequences are aligned "one to one".



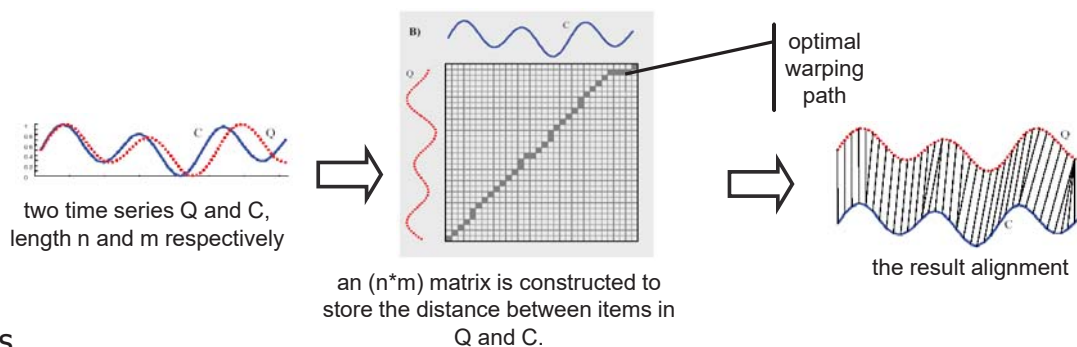
**"Warped" Time Axis**  
Nonlinear alignments are possible.



[M. Sridhar 07]

## ■ Principles

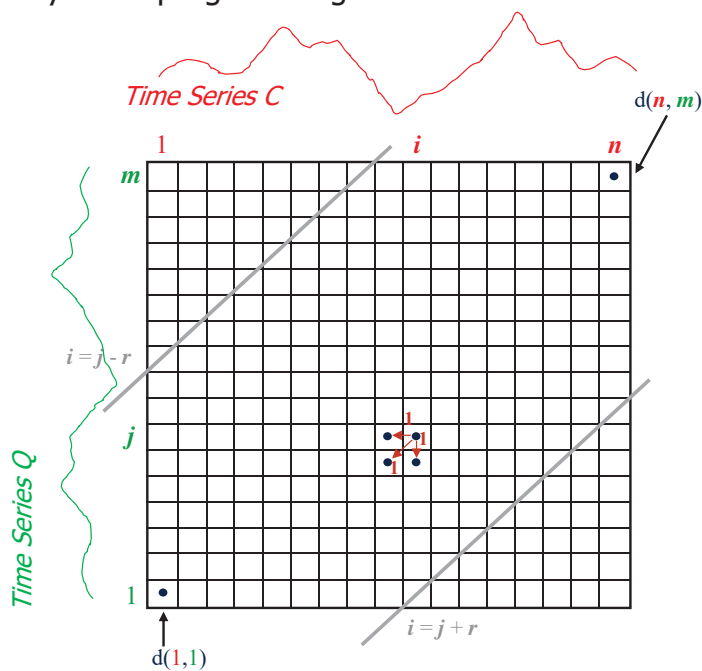
- Given: two sequences  $C: x_1, x_2, \dots, x_n$  and  $Q: y_1, y_2, \dots, y_m$
- Wanted: align two sequences base on a common time-axis



## ■ Conditions

- Boundary conditions: We want the path not to skip a part
  - Monotonicity: The alignment path does not go back in "time" index
  - Continuity: The alignment path does not jump in "time" index
- ... A good alignment path is unlikely to wander too far from the diagonal

## Dynamic programming



$d(i,j)$  = distance between  $Q_i$  &  $C_j$  - for instance  $(Q_i - C_j)^2$

$D(i,j)$  = distance cumulée

Initial condition:

$$D(1,1) = d(1,1)$$

$$D(1,j) = \sum_{p=1}^j d(1,p) \quad j = 1 \dots m$$

$$D(i,1) = \sum_{q=1}^i d(q,1) \quad i = 1 \dots n$$

DP-equation:

$$D(i,j) = \min \begin{pmatrix} D(i,j-1) \\ D(i-1,j-1) \\ D(i-1,j) \end{pmatrix} + d(i,j)$$

Warping window:  $j-r \leq i \leq j+r$ .

Time-normalized distance:

$$D(Q, C) = d(n, m) / c$$

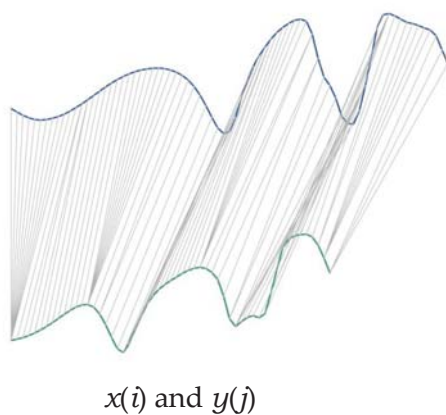
$$c = n + m.$$

The warping path

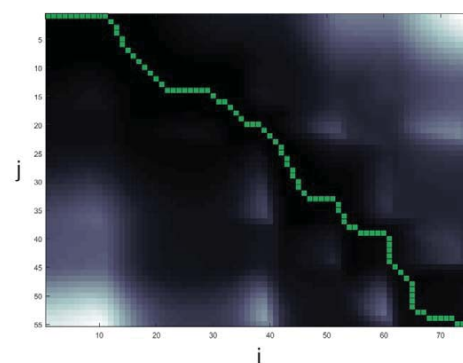
$$\varphi_{qp}(k) = (\varphi_{qp}^q(k), \varphi_{qp}^p(k))$$

## Alignment of two pairs of signals

- The matching between points of two pairs of signals
- Superimposition of the warping path ( $\varphi_{xy}$ ) on the cumulative distances matrix  $D$ .



$x(i)$  and  $y(j)$

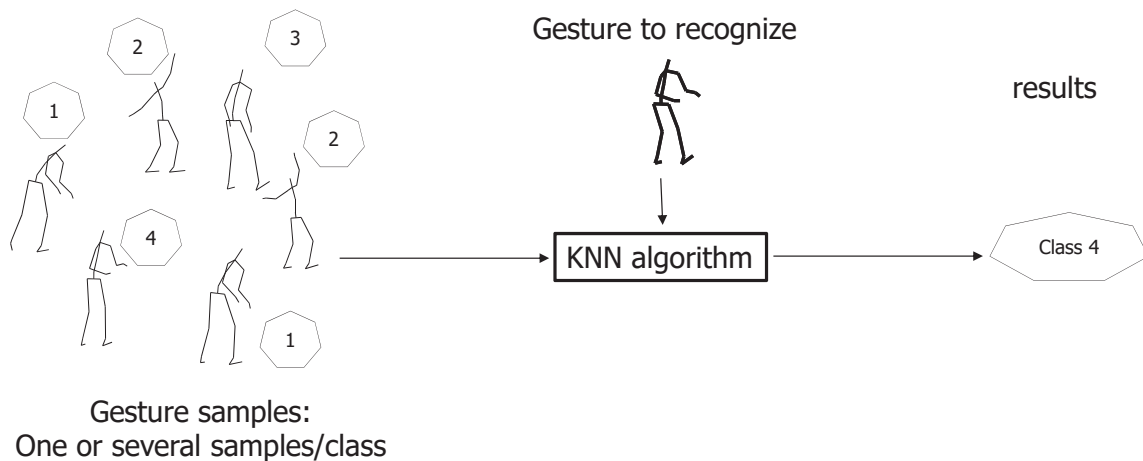


warping path ( $\varphi_{xy}$ )

[Morel, 2017]

■ General Principle:

- Classification: Distance-based methods
- ➔ K Nearest-Neighbor Classifier

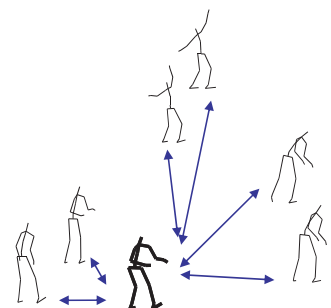


■ Basic idea

- Similarity can be described as distance in a specific space
  - We can use DTW for estimate the distance between tow sequences (gesture)
  - We can use a set of feature for estimate the distance between tow sequences (gesture)
- If suitable features were selected, that means
  - patterns of the same class have similar features
  - patterns of different classes have dissimilar features

■ Need to define

- A distance function  $d(x, y)$  for two arbitrary patterns  $x$  and  $y$



### ■ 1 Nearest-Neighbor Classifier (1NN)

- Assumption: for each pattern class  $C_i$ ,  $1 \leq i \leq N$  exactly one (representative) prototype  $Z_i$  is given.
- For an unknown pattern  $x$  the following classification rule is then valid:

$$k = \arg \min_i \{d(x, Z_i) | 1 \leq i \leq N\} \implies x \in C_k$$

### ■ Task

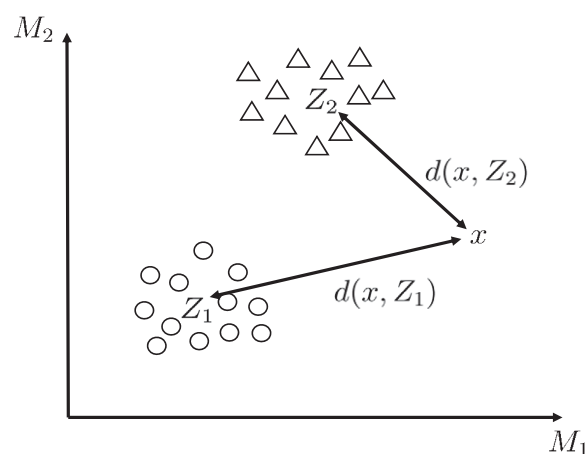
- Assign  $x$  to the class  $C_k$ , to which the next neighbor  $Z_k$  in the feature space belongs
- Reject  $x$ , if no unique minimum among  $d(x, Z_i)$  exists or if the existing unique minimum is too large

### ■ Nonparametric models

- requires storing and computing with the entire data set.

### ■ Nearest-Neighbor Classifier

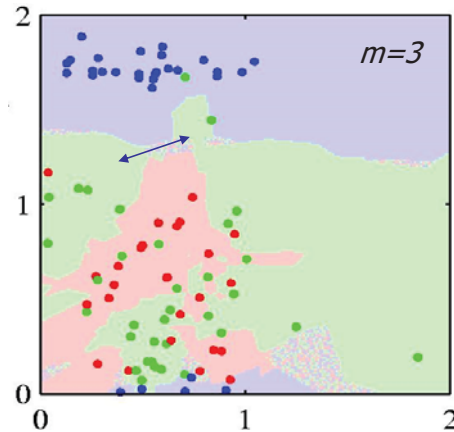
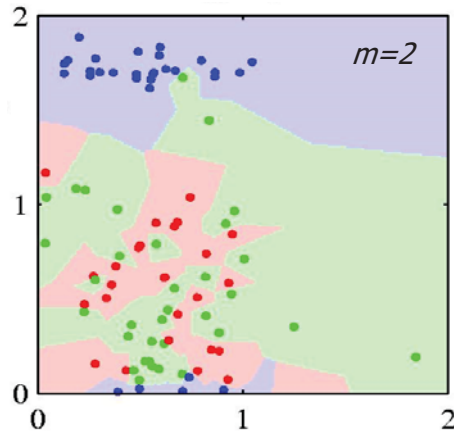
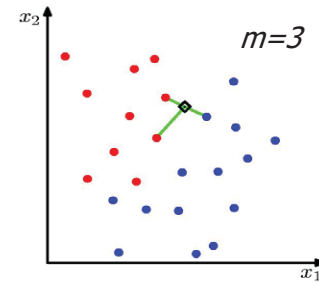
- Two pattern classes  $C_1$  and  $C_2$  in the two-dimensional feature space





### ■ k-Nearest-Neighbor Classifier

- Observe the  $m$  next neighbors of a pattern  $x$  from the sample set.
- Assign  $x$  to the class  $C_k$ , which occurs most frequently under all  $m$  next neighbors.
- Common selection  $3 \leq m \leq 7$



[Christopher M. Bishop]

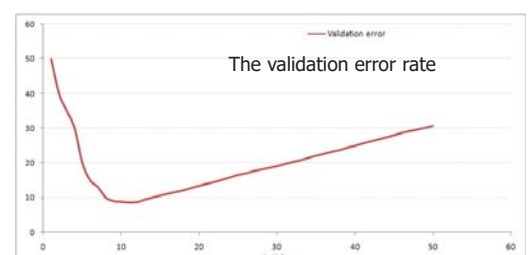
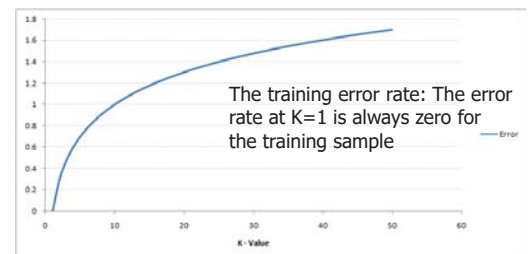
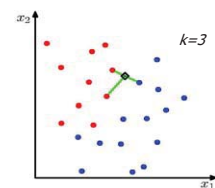
© eric.anquetil@irisa.fr

### ■ k-Nearest-Neighbor Classifier

- How to choose k
- Common selection  $3 \leq m \leq 7$

### ■ Define K by validation error rate

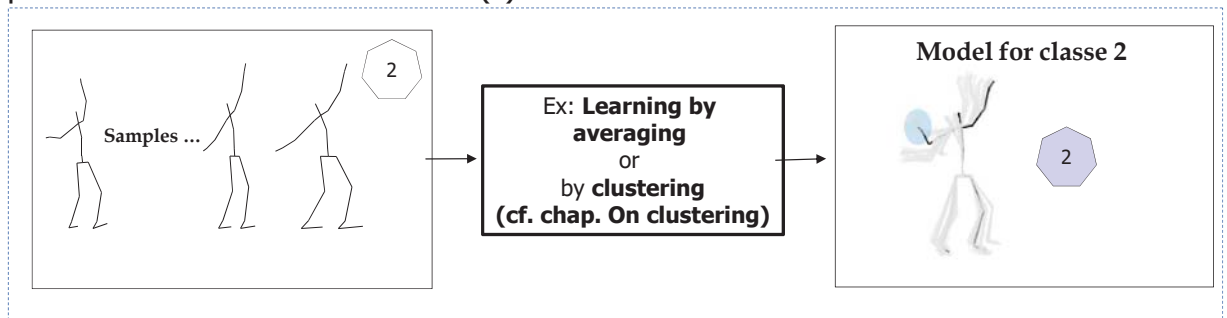
- Split the training and validation from the initial dataset.
- Plot the validation error curve to get the optimal value of K.
- This value of K should be used for all predictions.



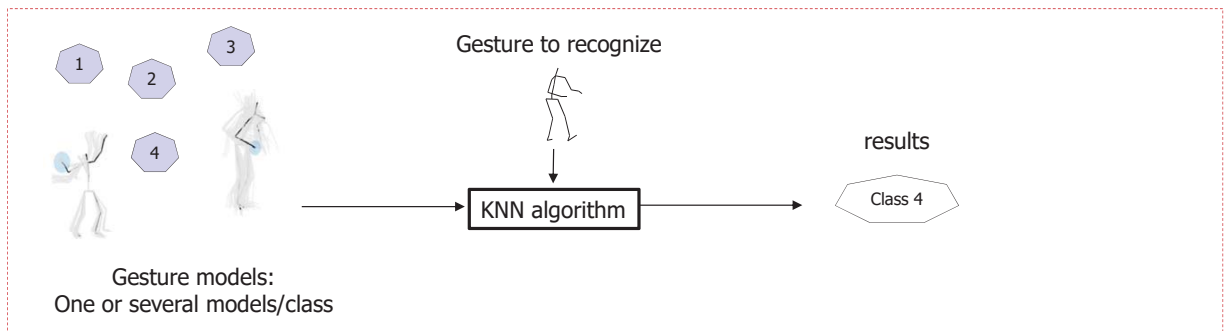
© eric.anquetil@irisa.fr

- A basic example to learn one or several model(s) for each class

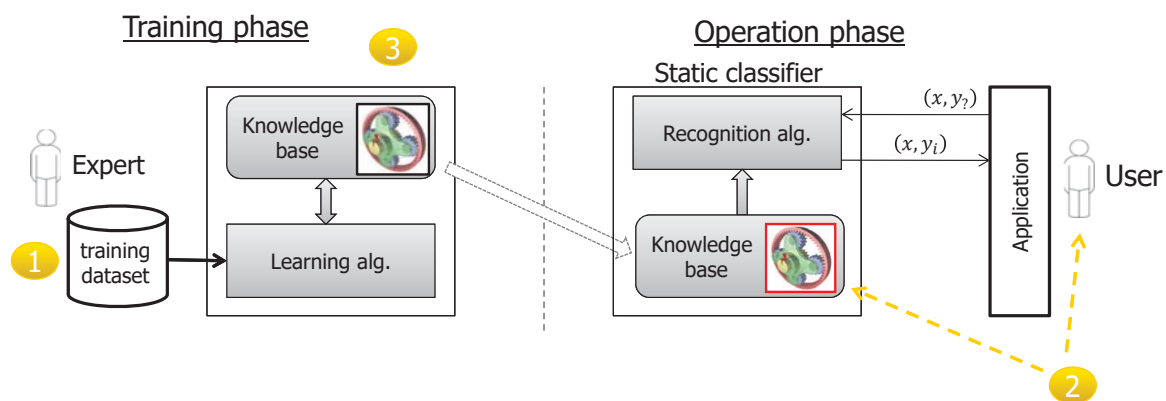
Learning phase  
for each class



Generalisation  
phase



[Almaksour 2011]

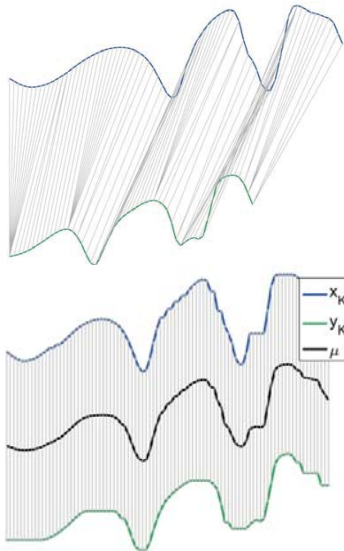


## • Limitations

- 1 Collecting large and exhaustive training dataset
- 2 User data can be much different from training data (different contexts/habits/needs, time-changing, ...)
- 3 Predefined and fixed set of categories/classes (included in the training dataset)

■ First idea to average to signals

- Alignment based on the warping path  $\phi_{xy}$  of length  $K$
- Creation of two new aligned signals  $x_K(k)$  and  $y_K(k)$  with the same length  $K$ 
  - $x_K(k) = x(\phi_{xy}(k))$        $y_K(k) = y(\phi_{xy}(k))$



$x(i)$  and  $y(j)$

$x_K(k)$  and  $y_K(k)$  with the same length  $K$   
 ➔ result from the resampling of signals  $x(i)$  and  $y(j)$  relatively to  $\phi_{xy}(k)$   
 ➔ average signal is  $\mu(k)$  (in black).

*On drawback: the average signal is longer than the two original signals*

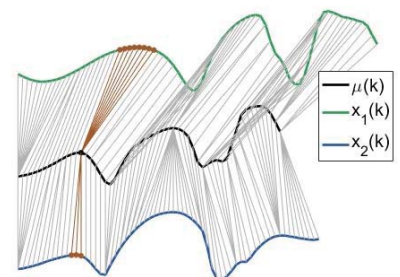
■ DTW Barycenter Averaging (DBA) [Petitjean et al. in 2011]

- A global averaging method for dynamic time warping.
- A fast algorithm that insures that the average **signal will have a reasonable length**.

■ The main steps of the algorithms:

- 1/ Randomly choose a signal  $x_0(k)$  from the dataset to initialize the average signal:  

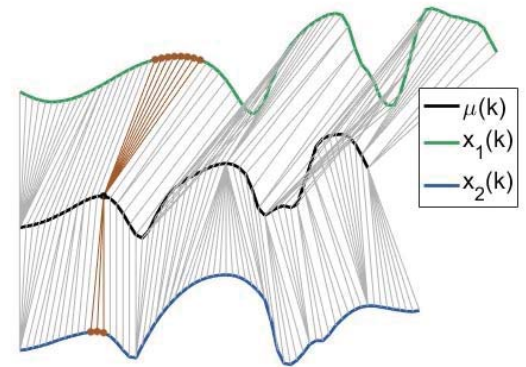
$$\mu(k) = x_0(k), \quad k = 1, \dots, M_0 \quad \text{where } M_0 \text{ is the length of } x_0(k).$$
- 2/ Iterate  $IT$  times the following steps:
  - (a) Align all signals  $x_i(k)$  on  $\mu(k)$  and compute warping paths  $\phi_{\mu x}$
  - (b) Update every point of the average signal  $\mu(k)$  as the **barycenter** of points associated to it during step (a).



**Algorithm 1** *DBA : averagingDTW*

**Require:**  $x_0(k)$  of length  $M_0$ ,  $(x_l(k))_{l=1\dots L}$  of lengths  $M_l$ ,  $IT$   
 $K = M_0$ ,  $\mu(k) \leftarrow x_0(k)$ ,  $k = 1, \dots, K$   
**for**  $it \in 1\dots IT$  **do**  
     $assocTab[k] = \emptyset$ ,  $k = 1\dots K$   
    **for**  $l \in 1\dots L$  **do**  
         $\varphi_{\mu x_l} \leftarrow DTW(\mu, x_l)$   
         $p \leftarrow length(\varphi_{\mu x_l})$   
        **while**  $p \geq 1$  **do**  
             $(k, n) \leftarrow \varphi_{\mu x_l}(p)$   
             $assocTab[k] \leftarrow assocTab[k] \cup \{x_l(n)\}$   
             $p \leftarrow p - 1$   
        **end while**  
    **end for**  
    **for**  $k \in 1\dots K$  **do**  
         $\mu(k) \leftarrow mean(assocTab[k])$   
    **end for**  
**end for**  
**return**  $\mu(k)$ ,  $k = 1, \dots, K$

[Petitjean *et al.* in 2011]  
 [Morel *et al.* in 2017]



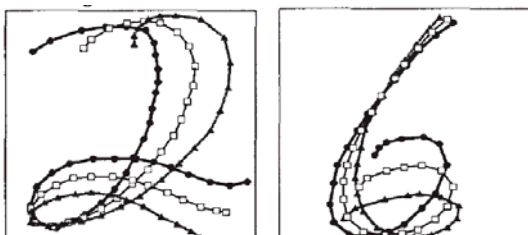
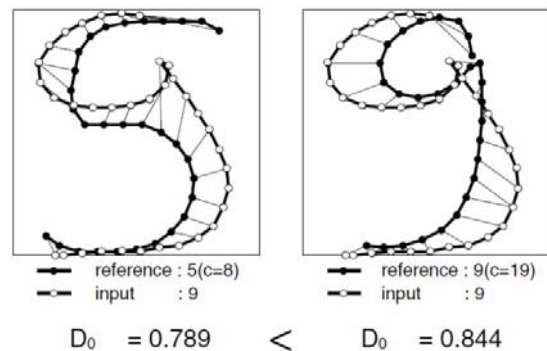
■ **Problematic**

- Misrecognitions due to overfitting

■ **Idea**

- To category specific deformations, called eigen-deformations, to suppress misrecognitions due to overfitting

[UCHIDA 2005, MOREL 2017]



## ■ Some results

- Estimating deformation tendencies
- Optimization based on DTW: learning geometric distortions from several examples of the same symbol [UCHIDA 2005]

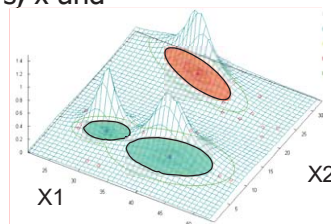
## ■ NB: Mahalanobis Distance

- Euclidean distance can be re-written as a dot-product operation

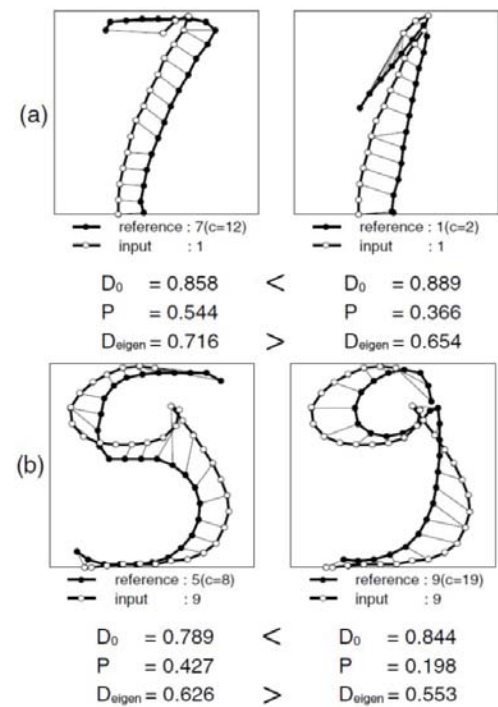
$$d_{L2}(x, y) = \sqrt{(x - y)^T (x - y)}$$

- Mahalanobis distance between two vectors,  $x$  and  $y$ , where  $S$  is the covariance matrix.

$$d_M(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)}$$



[UCHIDA 2005]

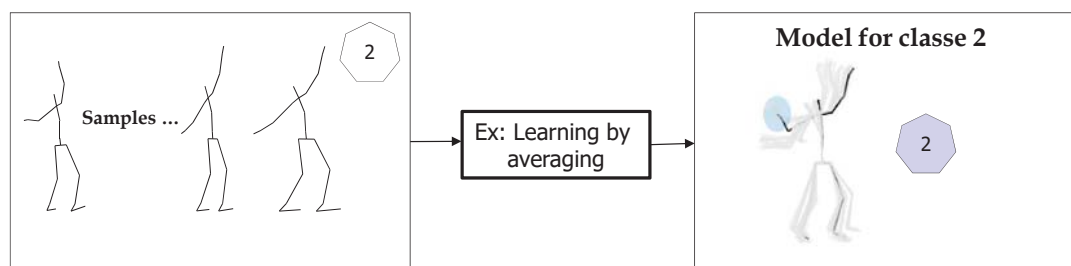


© eric.anquetil@irisa.fr

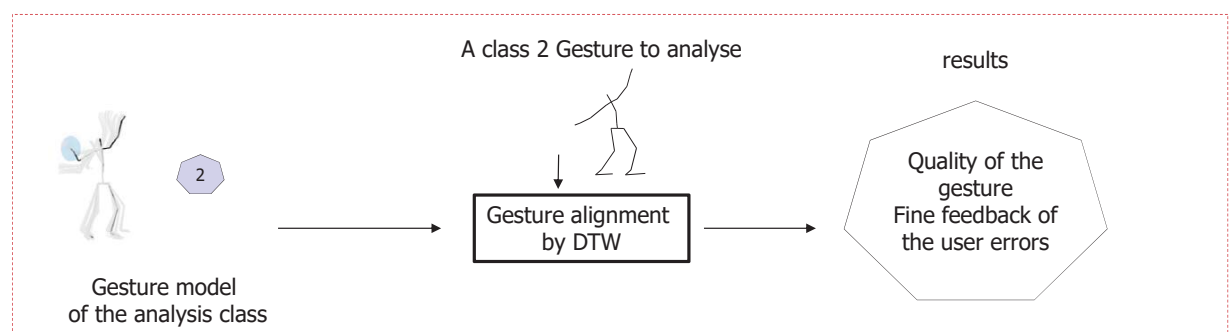
## \_Chap. 8 | DTW : For Gesture Analysis

- DTW can also be used for fine gesture analysis (virtual sportive coaching)

Learning phase  
for each class



Gesture  
analysis phase



© eric.anquetil@irisa.fr

- Levenshtein distance
  - Insertion
  - Deletion
  - Substitution
- Extension
  - Fusion
  - Division
  - Pair substitution

**Entrée:**  
 $X = x_1x_2...x_n$  : une chaîne de caractères  
 $Y = y_1y_2...y_m$  : une chaîne de caractères  
 $d$  : une matrice de taille  $|X| + 1 \times |Y| + 1$  permettant de stocker les résultats intermédiaires

**Initialisation**  $d(0,0) = 0$   
**Pour**  $i$  de 1 à  $n$  **Faire**  
     -  $d(i,0) = d(i-1,0) + 1$   
**Fin Pour**  
**Pour**  $j$  de 1 à  $m$  **Faire**  
     -  $d(0,j) = d(0,j-1) + 1$   
**Fin Pour**  
**Pour**  $i$  de 1 à  $n$  **Faire**  
     **Pour**  $j$  de 1 à  $m$  **Faire**  
         **Si**  $x_i = y_j$  **Alors**  
             -  $d(i,j) = d(i-1,j-1)$   
         **Sinon**  
             -  
         **Fin Si**  
     **Fin Pour**  
**Fin Pour**  
**Sortie:**  $d(n,m)$

→ 

nage	→	nuage
nuage	→	nage
nage	→	page

→ 

clé	→	dé	clé
aib	→	aile	aib
méanche	→	méandre	méandre

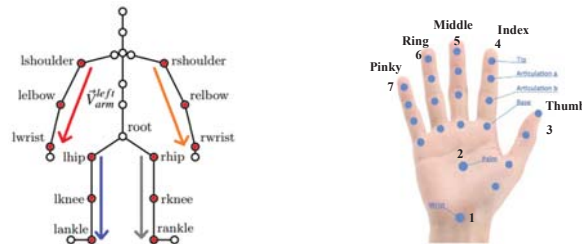
$$d(i,j) = \min \begin{cases} d(i-1,j-1) + 1 & \text{Coût substitution} \\ d(i-1,j) + 1 & \text{Coût suppression} \\ d(i,j-1) + 1 & \text{Coût insertion} \end{cases}$$

ic.anquetil@irisa.fr

## \_Chapitre 9

### Pre-segmented Action Recognition: Skeleton based and "Statistical" approaches

- A pattern refers to either a **whole body action** or **dynamic hand gesture**

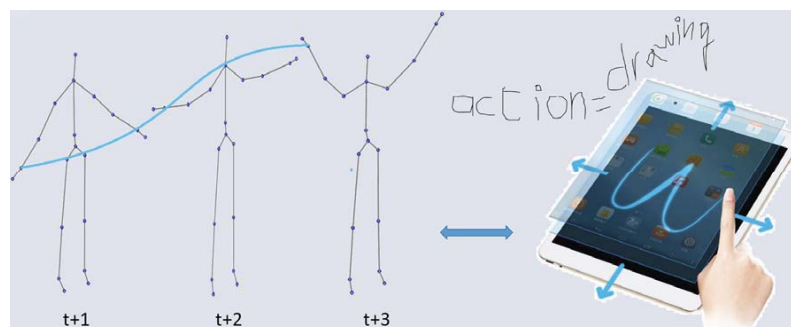


- The overall process for segmented pattern representation and recognition is:



## \_Chap. 9 | Skeleton based Action Recognition based on 3D gesture trajectories

- Addressing 3D action recognition in light of 2D representation
  - 3D gesture trajectories may be processed similarly to hand-drawn trajectories
    - Same data type (trajectories or signal)
  - Graphonomic characteristic:
    - a human is the performer
    - Well-established 2D experience





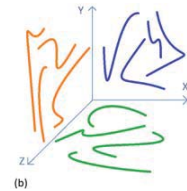
- Pre-segmented Action Recognition:  
Skeleton based and « statistical » approaches (using SVM)

- Example of two approaches [Boulahia 2017]:

- A first naïve approach:

- **3DMM** : 3D Multistroke Mapping

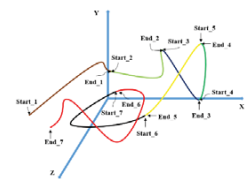
- *3D Multistroke Mapping (3DMM): Transfer of hand-drawn pattern representation for skeleton-based gesture recognition. In 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), 2017.*



- A more robust approach:

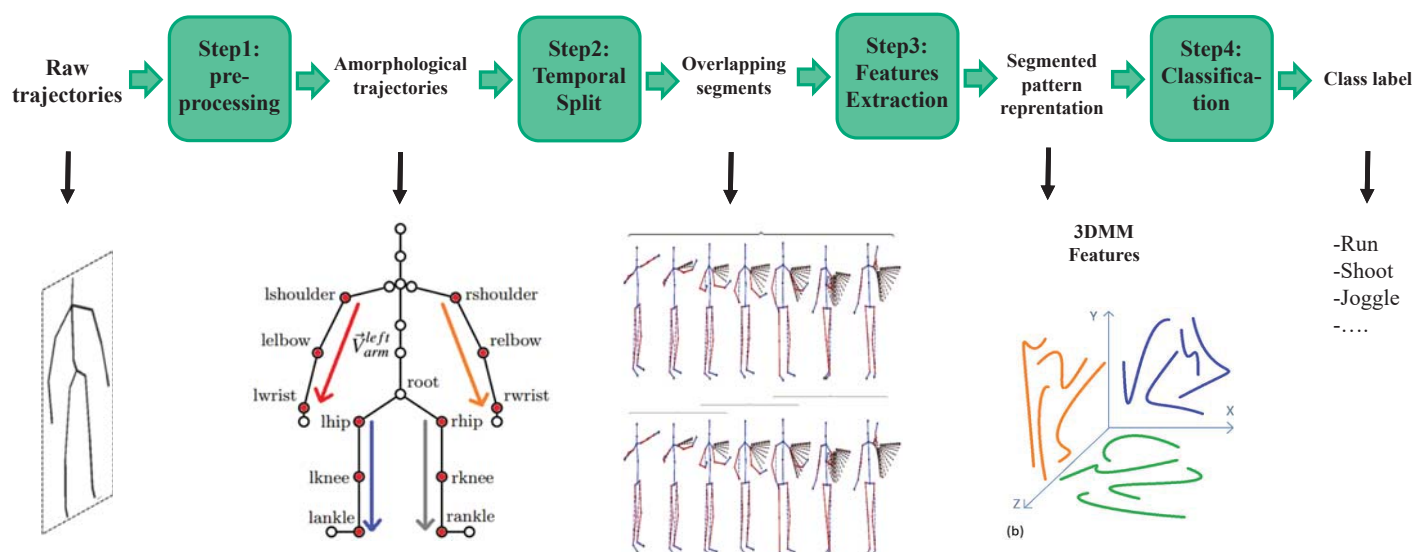
- **HIF3D**: Handwriting-Inspired Features for 3D action recognition

- *HIF3D: Handwriting-Inspired Features for 3D skeleton-based action recognition. In 23rd IEEE International Conference on Pattern Recognition (ICPR), 2016.*



© eric.anquetil@irisa.fr

- The overall process for segmented action representation and recognition is:

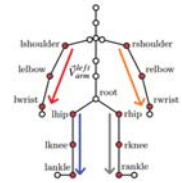


© eric.anquetil@irisa.fr



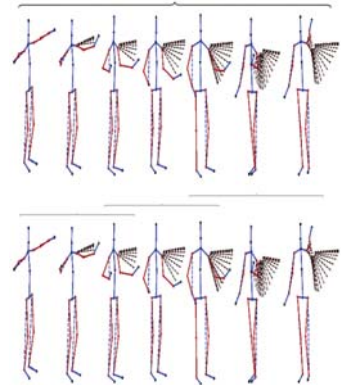
### ■ Step 1: Pre-processing

- Goal: address the morphological variability issue
- How: perform a normalisation of the raw trajectories according to the subject morphology



### ■ Step 2: Temporal split

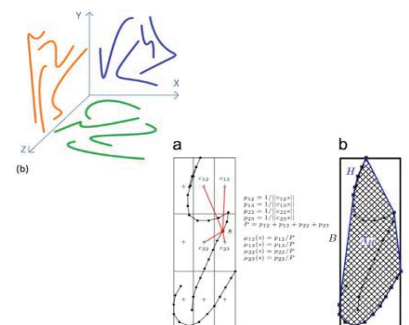
- Goal: address the morphological sequencing issue (for instance if two arms are raised at the same time or one after another, the model should distinguish these two different patterns)
- How: Extract partial segments from the whole pattern according to overlapping sliding windows



© eric.anquetil@irisa.fr

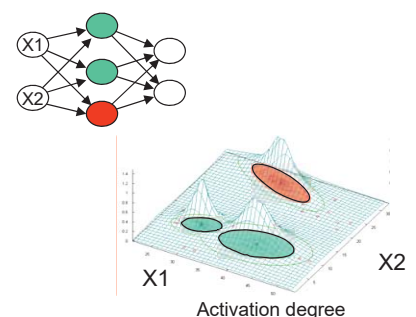
### ■ Step 3: Features extraction

- Goal: build the pattern representation that should get the spatial relationship between trajectories and the overall shape of the produced pattern
- How: It consists in extracting a set of features on the whole pattern and on the overlapping segments produced in step 2



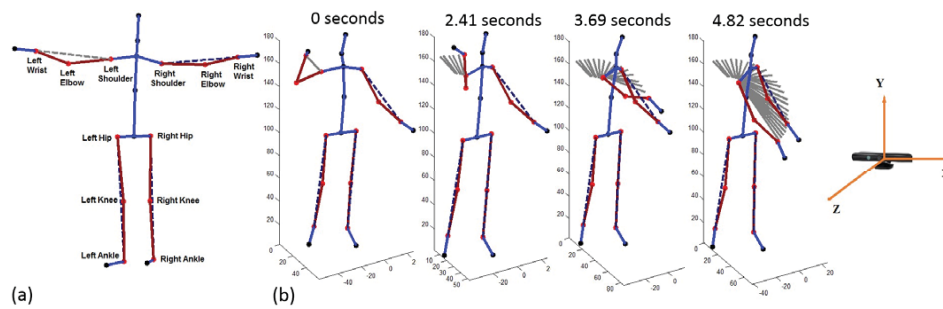
### ■ Step 4: Classification

- Goal: get the class label
- How: using a classifier (here **SVM** or **MLP**) trained on a training set and then applied on each testing pattern



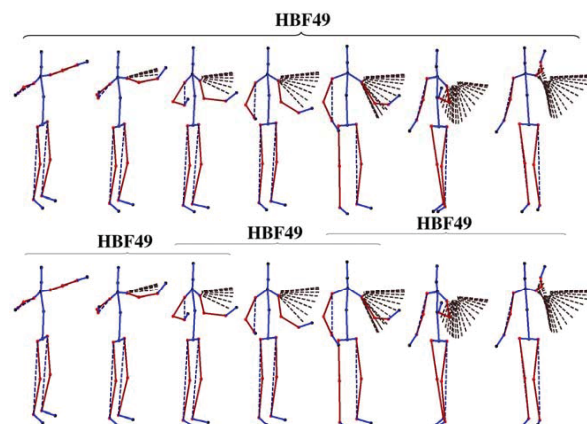
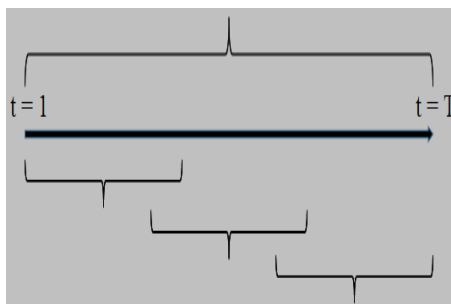
© eric.anquetil@irisa.fr

■ Addressing morphological variability before trajectory extraction



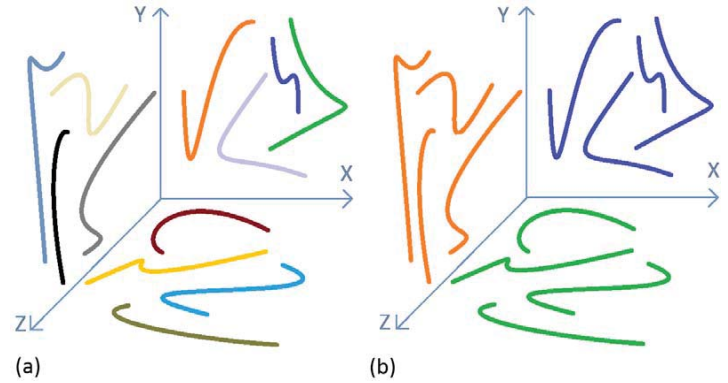
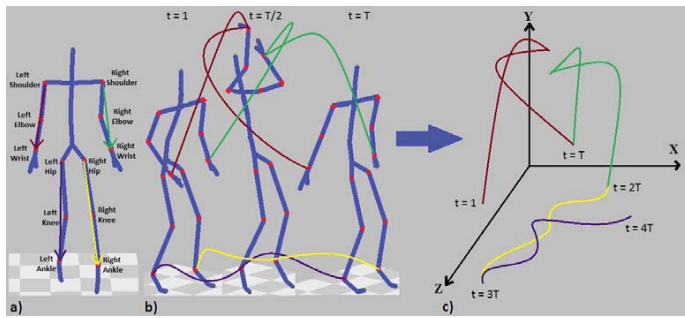
[Kulpa 2005] "Morphology-independent representation of motions for interactive human-like animation", 2005.

- Modelling temporal information: Temporal Split Extraction
  - Handling temporal sequencing
  - Features are extracted according to two temporal levels (Level = 2)
- Number of features:
  - Without selection :  $4 \times 49 \times 3 = 588$
  - With selection: between 400 and 80



## \_Chap. 9 | Action representation by 3DMM: Step 3 - : dealing with the set of 2D trajectories

- A first naïve approach 3DMM using direct 2D projection [Boulahia 2016]
- Several strategies to consider all the trajectories
  - (a) Mono-Stroke approach
    - We loss the spatial dependencies
  - (b) Multi-strokes approach
    - Modelling spatial relationship



## \_Chap. 9 | Action representation by 3DMM: Step 3 - Direct 2D features extraction

- [Delaye and Anquetil] "HBF49 feature set: A first unified baseline for online symbol recognition", 2013.

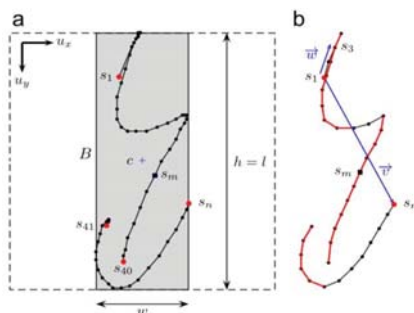


Figure: Descripteurs dynamiques  
(positions de départ, longueur des strokes, inflexion)

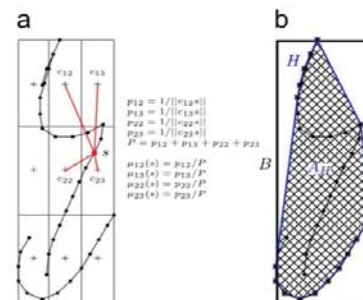
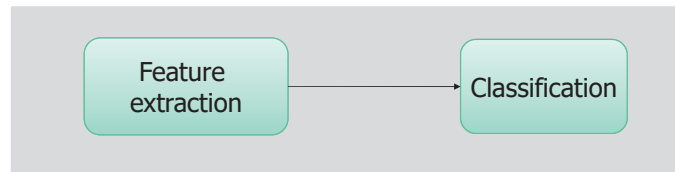


Figure: Descripteurs statiques  
(histogramme 2D, boîte englobante)

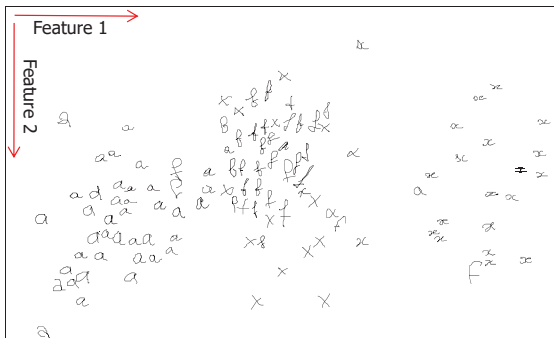
## Classification



Feature extraction  
Discriminative features

Classification  
Use a feature vector  
to assign the object to a category (class)

Here, 2 dimensions Feature space



Here, discrimination of 3 classes: "a", "f", "x"

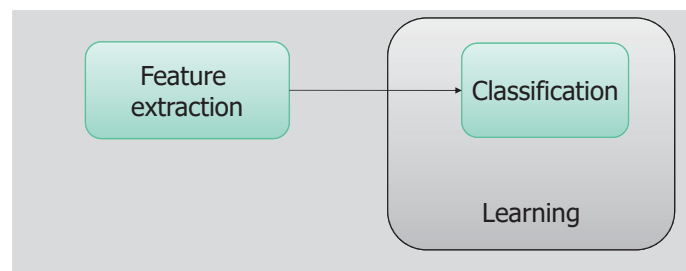


(decision boundary)

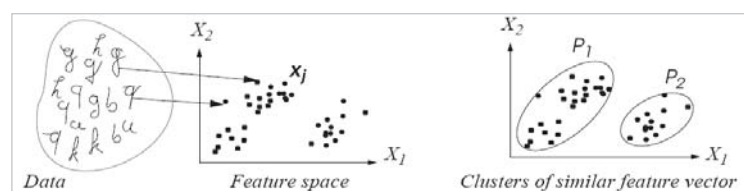
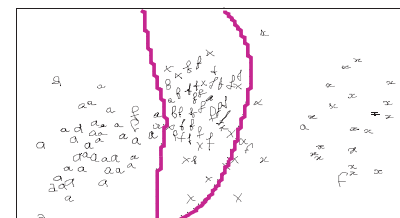
© eric.anquetil@irisa.fr

## Learning

- Finding all the parameters of a classifier based on a training set.



- Supervised learning: Generalization
  - For the learning, a teacher provides a category/class label for each pattern in the training set
- Unsupervised learning: Clustering
  - The system forms clusters or "natural groupings" of the input patterns

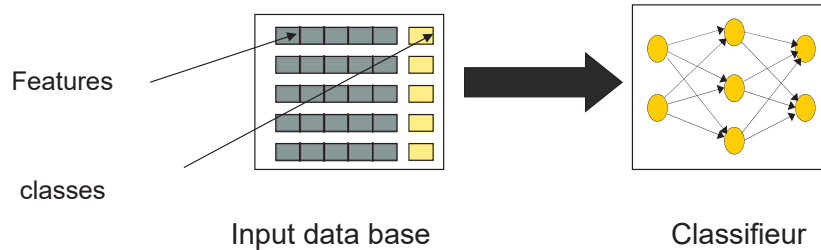


© eric.anquetil@irisa.fr

## ■ Learning and generalization capacities

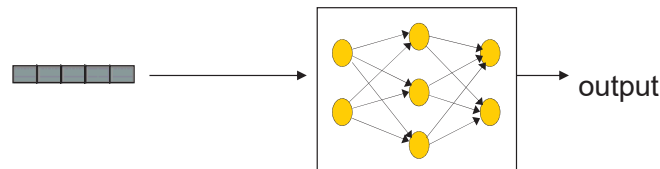
### ■ Learning

- consists of presenting an input pattern and modifying the network parameters (weights) to reduce distances between the computed output and the desired output



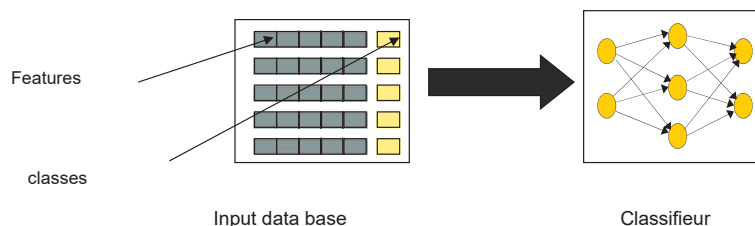
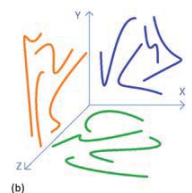
### ■ Generalization / Feedforward

- consists of presenting a pattern to the input units and passing the signals through the network in order to get outputs units



## ■ Learning: Number of features

- For each temporal windows: 49 features [HBF 49] x 3 projections = 147
- 4 temporal windows: the total length of features
  - 588 ( $147 \times 4$ ).
- Feature selection:
  - To limit redundancy
  - between 400 and 80



### ■ HDM05 dataset

- HDM05 is an optical marker-based dataset
  - M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, A. Weber: **Documentation Mocap Database HDM05**. Technical report, No. CG-2007-2, ISSN 1610-8892, Universität Bonn, June 2007.
- Contains around **100 motion classes** including
  - various walking and kicking motions, cartwheels, jumping jacks, grabbing and depositing motions, squatting motions and so on.
- Each motion class contains **10 to 50 different instances** of the same type of motion

### ■ Experimental Protocol

- Evaluation with **11 motion actions**.
- The actions are performed by **5 subjects**, while each subject performs each action a couple of times ;
  - this suggests a set of **249** sequences.

### ■ Testing protocol

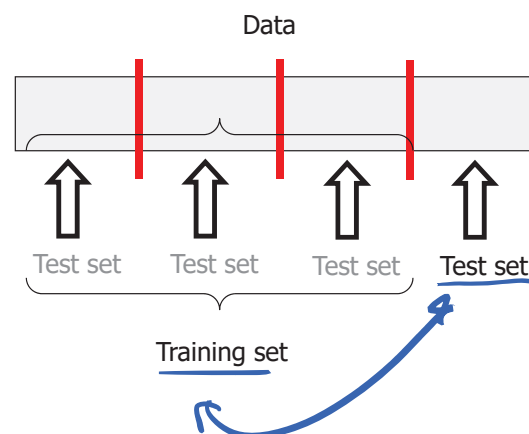
- **3** subjects for learning (142 instances)
- **2** subjects for testing (109 instances)
- **cross-subjects validation**



© eric.anquetil@irisa.fr

### ■ Cross-Validation: K-fold

- Successively setting apart a block of data (instead of a single observation)



© eric.anquetil@irisa.fr

■ Results (HDM05 dataset)

Method	Authors & Year	#Features	Reco. rate (%)
Dynamic Time Warping	[Reyes et al., 2011]	-	82.08
MIJA/MIRM + LCSS	[Pazhoumand-Dar et al., 2015]	-	85.23
SMIJ + Nearest neighbour	[Ofli et al., 2014]	-	91.53
LDS + SVM	[Chaudhry et al., 2013]	-	91.74
Skeletal Quads + SVM	[Evangelidis et al., 2014]	9360	93.89
Cov3DJ + SVM	[Hussein et al., 2013]	43710	95.41
BIPOD + SVM	[Zhang and Parker, 2015]	-	96.70
HOD + SVM	[Gowayyed et al., 2013]	1116	97.27
<b>3DMM + SVM + Level = 1</b>		100	<b>91.74</b>
<b>3DMM + MLP + Level = 1</b>		20	<b>92.66</b>
<b>3DMM + SVM + Level = 2</b>		400	<b>94.49</b>
<b>3DMM + MLP + Level = 2</b>		80	<b>94.49</b>

Table: Comparisons between **3DMM** approach, with and without temporal split, and previous approaches on the **HDM05** dataset.

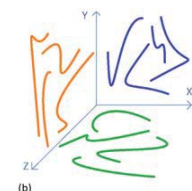
■ Pre-segmented Action Recognition:  
Skeleton based and « statistical » approaches (using SVM)

■ Example of two approaches [Boulahia 2017]:

■ A first naïve approach:

■ **3DMM** : 3D Multistroke Mapping

- *3D Multistroke Mapping (3DMM): Transfer of hand-drawn pattern representation for skeleton-based gesture recognition. In 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), 2017.*



■ A more robust approach:

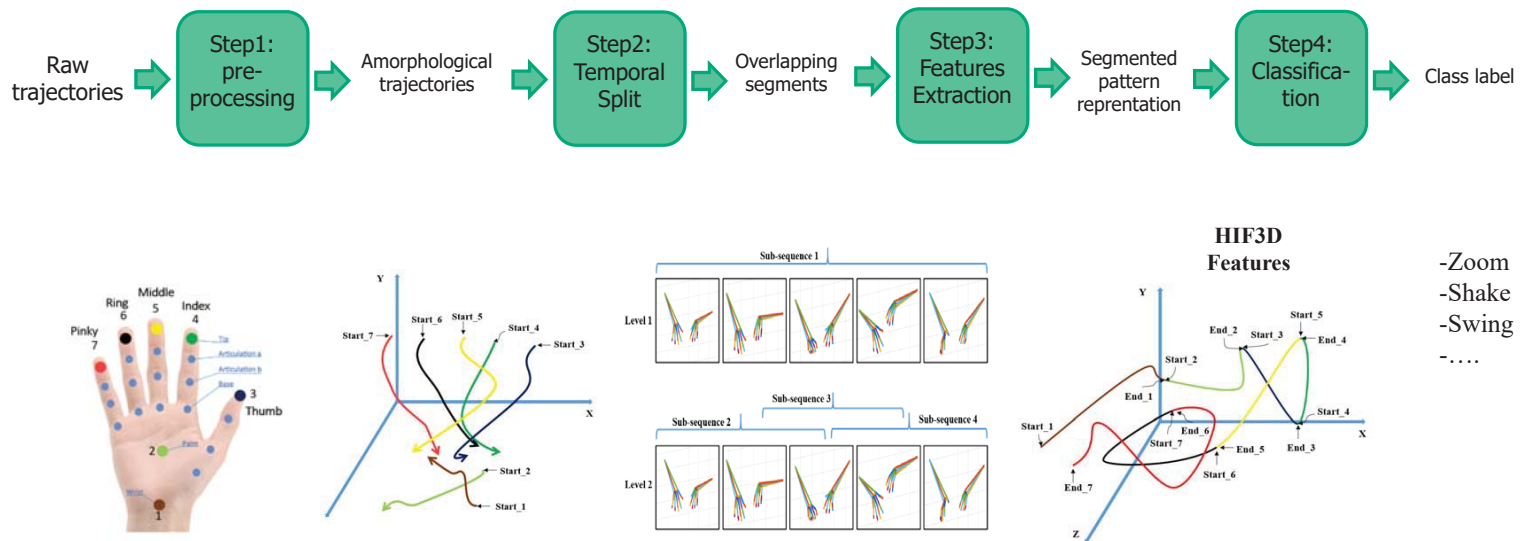
■ **HIF3D**: Handwriting-Inspired Features for 3D action recognition

- *HIF3D: Handwriting-Inspired Features for 3D skeleton-based action recognition. In 23rd IEEE International Conference on Pattern Recognition (ICPR), 2016.*





- The overall process for segmented dynamic hand gesture recognition:



- Overview of the features
  - A new feature-set inspired by an efficient hand-drawn descriptor but **entirely dedicated to the 3D skeleton trajectories**
  - HIF3D**: Handwriting-Inspired Features for 3D skeleton-based action recognition. [Boulahia, ICPR 2016].
    - Extending HBF49** to form HIF3D so as to process directly 3D trajectories instead of projecting
    - Better capturing the **correlation** between joint **trajectories**
    - Reducing dimensionality** and avoiding **redundancy**
    - Adding new features** (such as volume related features) which are more adapted to 3D patterns
- A set of **89** features (very compact comparing to existing feature-set)
  - 41 Extended features**, i.e. features which can directly be extended from 2D trajectory to 3D one.
  - 48 Newly features**, i.e. carry the characteristic information identified for handwritten pattern but have different formulations since the original 2D formulas can not be directly applied for the 3D case.



■ Extended features:

■ **Starting points:**  $f_1 = \frac{x_1 - c_x}{l} + \frac{1}{2}$ ,  $f_2 = \frac{y_1 - c_y}{l} + \frac{1}{2}$ ,  $f_3 = \frac{z_1 - c_z}{l} + \frac{1}{2}$

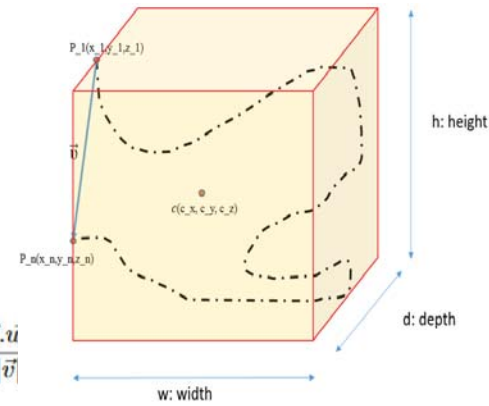
- $x_1, y_1$  and  $z_1$  are the coordinates of the first point of the pattern
- $c_x, c_y$  and  $c_z$  are the coordinates of the the center of the bounding box B
- $l$  is the greatest side of the bounding box B
- The bounding box B is the cuboid that enclose the pattern

■ **First point to last point vector:**  $f_7 = \|\vec{v}\|$ ,  $f_8 = \frac{\vec{v} \cdot \vec{u}_x}{\|\vec{v}\|}$ ,  $f_9 = \frac{\vec{v} \cdot \vec{u}_y}{\|\vec{v}\|}$ ,  $f_{10} = \frac{\vec{v} \cdot \vec{u}}{\|\vec{v}\|}$

- $v$  is the vector that relates the first and the last point of the pattern

■ **Bounding box diagonal angles:**  $f_{21} = \arctan\left(\frac{h}{w}\right)$ ,  $f_{22} = \arctan\left(\frac{d}{h}\right)$ ,  $f_{23} = \arctan\left(\frac{w}{d}\right)$

- $h, w$  and  $d$  are the height, the width and the depth of the bounding box B, respectively.



■ Newly features:

■ **3D zoning histogram:**

- We define a regular 3D partition of the bounding box B into  $3 \times 3 \times 3$  voxels resulting in twenty-seven zoning features
- **Histograms** are built by computing a fuzzy weighted contribution from each point  $s_i$  to its eight neighbouring voxels, where the weights are proportional to the distance from the point to the voxels center  $c_{j,k,l}$ .

$$f_{58} = \frac{1}{n} \sum_{i=1}^n \mu_{111}(s_i), \dots, f_{84} = \frac{1}{n} \sum_{i=1}^n \mu_{333}(s_i)$$

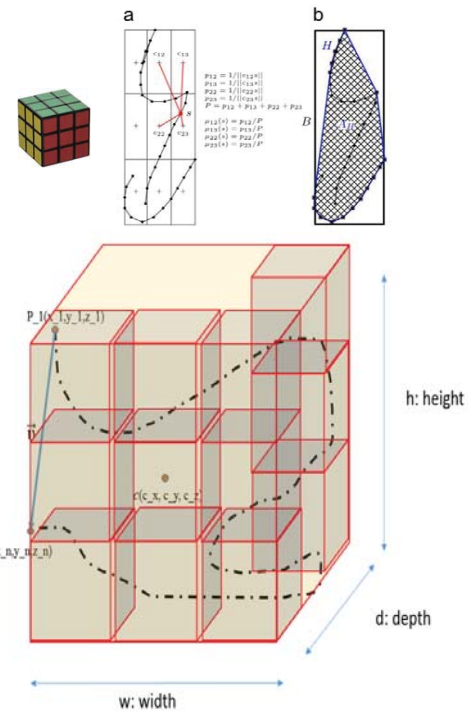
- With  $0 \leq \mu_{jkl}(s_i) \leq 1$  is the contribution of point  $s_i$  to the voxel with center  $c_{j,k,l}$  for each  $1 \leq j,k,l \leq 3$

■ **Convex Hull features:**

- To capture the overall shape produced during the gesture we consider the convex hull H of the resulting pattern S
- We first compute its convex hull volume  $V_H$ .
- Then we extract the normalized volume and the compactness as two additional features

$$f_{88} = \frac{V_H}{w * h * d}, \quad f_{89} = \frac{L^3}{V_H}$$

- $L$  is the total length of the pattern and  $w, h$  and  $d$  are the height, the width and the depth of the bounding box B, respectively



- Experimental Protocol : 3 subjects for learning (142 instances) + 2 subjects for testing (109 instances)

Method	Authors & Year	#Features	Reco. rate (%)
Dynamic Time Warping	[Reyes et al., 2011]	-	82.08
MIJA/MIRM + LCSS	[Pazhoumand-Dar et al., 2015]	-	85.23
SMIJ + Nearest neighbour	[Ofli et al., 2014]	-	91.53
LDS + SVM	[Chaudhry et al., 2013]	-	91.74
Skeletal Quads + SVM	[Evangelidis et al., 2014]	9360	93.89
Cov3DJ + SVM	[Hussein et al., 2013]	43710	95.41
BIPOD + SVM	[Zhang and Parker, 2015]	-	96.70
HOD + SVM	[Gowayyed et al., 2013]	1116	97.27
<b>3DMM + SVM + Level = 2</b>		400	94.49
<b>HIF3D + SVM + Level = 2</b>		356	<b>98.17</b>



Table: Comparisons between **HIF3D** approach, with temporal split, and previous approaches on the HDM05 dataset.

## \_Chapitre 10

### Non-segmented Action Recognition: Skeleton based and "Statistical" approaches

❖ **Introduction: understand the problematic of gesture interaction**

- What is a gesture: the different natures of gestures
- Human Computer Interaction: new opportunities

❖ **Gesture recognition: Isolated Gestures Classification (segmented)**

- Overview of the task: recognizing isolated gestures (The overall pattern recognition process)
- Machine Learning and Pattern recognition: a short overview of some existing techniques
  - Gesture classification: "Time-series" approaches
  - Pre-segmented Action Recognition: Skeleton based and "Statistical" approaches

❖ **Gesture recognition in real-time streaming (non segmented)**

- Overview of the task: recognizing in real-time streaming
- Non-segmented Action Recognition: Example of one approaches [Boulahia 2017]
- Presentation of experimental results using Kinect and Leap Motion

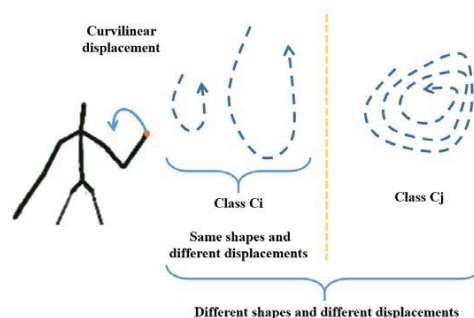
❖ **Early Gesture recognition**

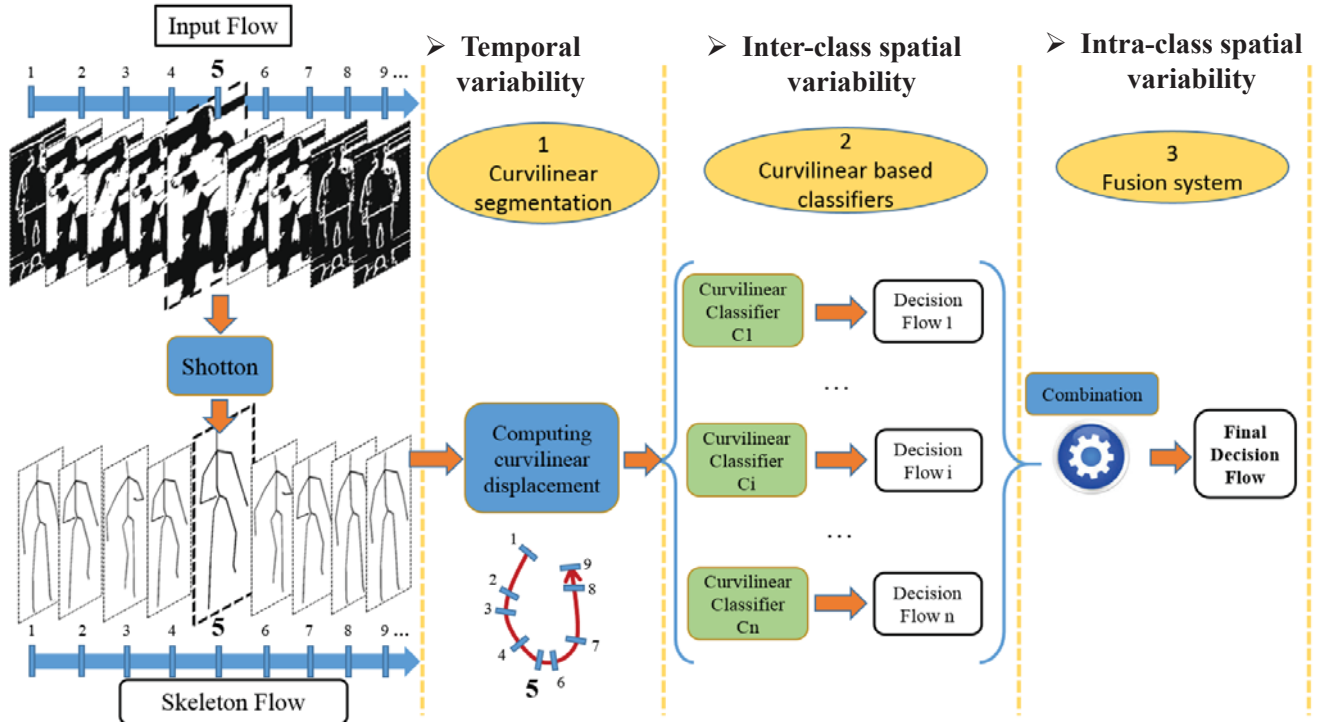
101

102

■ The challenges that should be addressed are:

- Temporal variability: that occurs when subjects perform gestures with different speeds.\*
- Inter-class spatial variability: which refers to disparities between the displacement amounts induced by different classes (i.e. long vs. short movements).
- Intra-class spatial variability: caused by differences in style and gesture amplitude.





### Step 1: curvilinear segmentation

- Dynamically defining windows depending on the amount of information (i.e. motion) available in the unsegmented flow.
  - The metric used to measure the amount of information is the curvilinear displacement of joints.

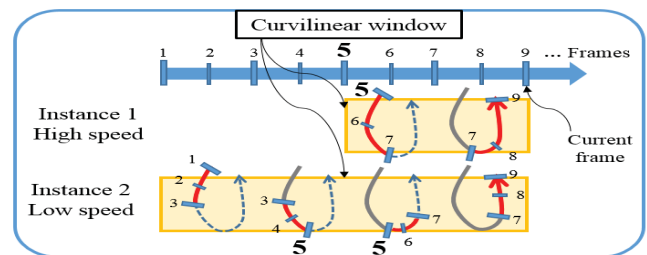
- function  $CuDi(F_S, F_E)$  that computes the **curvilinear displacement** for a given motion segment, starting at frame  $F_S$  and ending at  $F_E$ , as follows:

$$CuDi(F_S, F_E) = \sum_{i=F_S}^{i=F_E} d_i^{Avg}$$

- where  $d_i^{Avg}$  is the instantaneous average displacement

- **Curvilinear window** as being a sliding window

- whose size is continuously updated such that it encompasses, at each frame, a specific curvilinear displacement.



## Step 1: curvilinear segmentation

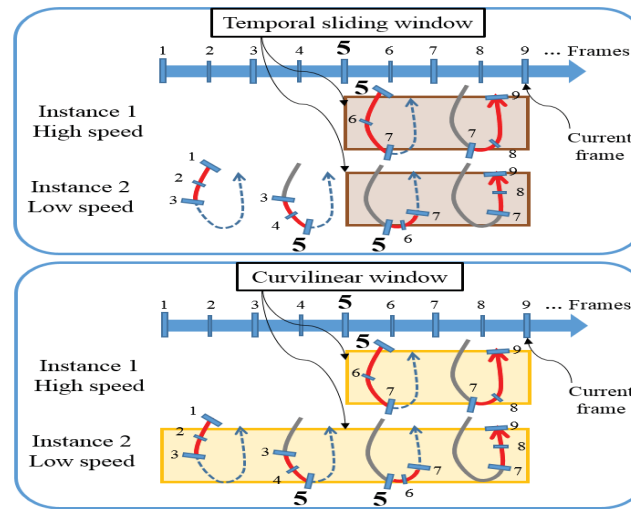
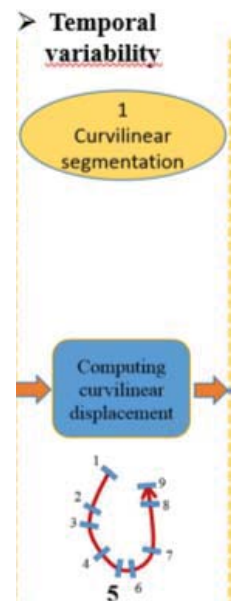


Illustration of the difference between the curvilinear window and the usual temporal sliding window.



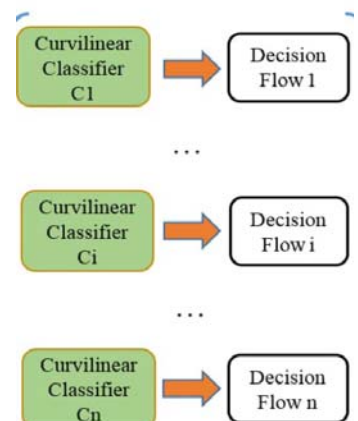
© eric.anquetil@irisa.fr

## Step 2: curvilinear-based classifiers

- To address the second issue, inter-class spatial variability, we propose to use as many classifiers as there are curvilinear displacements.
- Each classifier  $C_i$**  is trained to recognize all action classes but according to the **curvilinear size of classe  $G_i$**
- We constitute the training set of a classifier  $C_i$  by extracting local features (**HIF3D**) according to its corresponding curvilinear window.
- SVM** classifiers are then trained on each training set.

### Inter-class spatial variability

#### 2 Curvilinear based classifiers



© eric.anquetil@irisa.fr

### Step 3: Decision process (at each frame)

- The fusion system is mainly composed of:
  - as many **local histograms** as there are classifiers && a **global histogram**

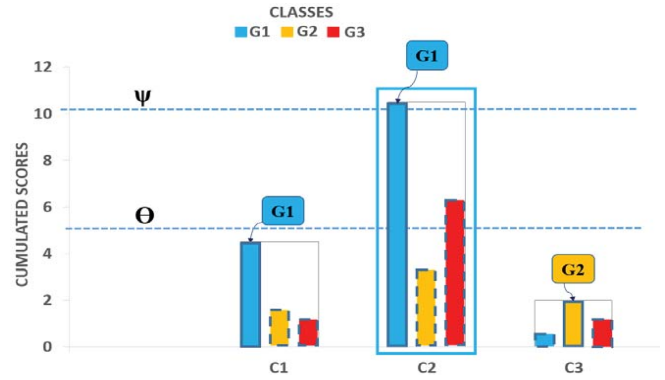
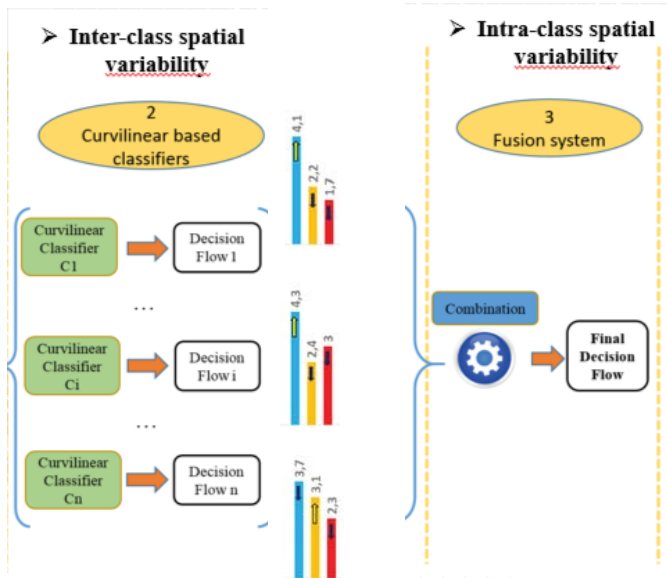


Illustration of the **global histogram** functioning at frame i with three classifiers which can G1, G2 or G3

© eric.anquetil@irisa.fr

### Step 3: Decision process

- Each local histogram has as many entries as there are classes to predict.
  - It is used to cumulate (at each frame) the score of each class predicted by the associated classifier  $C_i$ .
- Then, at each instant, each **local histogram** is updated
- the  $j$ th entry of a histogram  $H_{is_i}$  associated with classifier  $C_i$  is updated at each instant:
  - $\beta$  equals to the difference between
    - the score of the currently predicted class, i.e.  $Predicted\_i$ ,
    - and the score of the secondly ranked predicted class by the classifier  $C_i$ .
  - $\gamma$  corresponds to the difference between
    - the score of  $Predicted\_i$
    - and that of  $j$ th class corresponding to the  $j$ th entry of the histogram.

$$H_{is_i}(j) = \begin{cases} H_{is_i}(j) + \beta, & \text{if } j = Predicted\_i \\ H_{is_i}(j) - \gamma, & \text{otherwise} \end{cases}$$

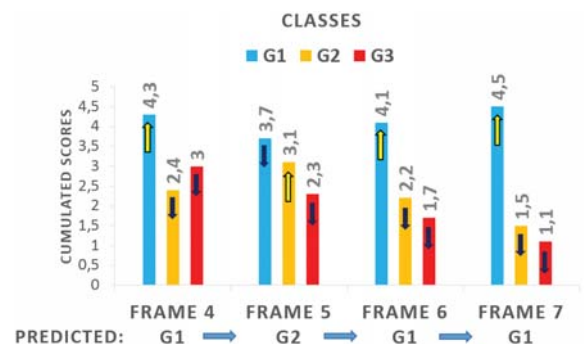


Illustration of a **local histogram** functioning with three classes at frames 4, 5, 6 and 7

© eric.anquetil@irisa.fr



### Step 3: Decision process

- Then, at each instant, each local histogram is used to update the global histogram.
  - This latter is responsible for emitting the final decision.
- At each decision, all histograms are reinitialized to zeros, as are the cumulated curvilinear displacements for each classifier.

$$Output = \begin{cases} G_i & , \text{ if } \exists 1 \leq i \leq n \text{ \& } His\_Global(i) \geq \theta \text{ \& } Output\_i = G_i \\ G_j & , \text{ if } \exists 1 \leq i \neq j \leq n \text{ \& } His\_Global(i) \geq \psi \text{ \& } Output\_i = G_j \\ ? & , \text{ otherwise} \end{cases}$$

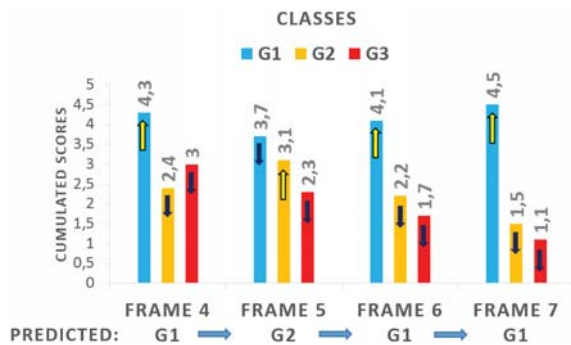


Illustration of a **local histogram** functioning with three classes at frames 4, 5, 6 and 7

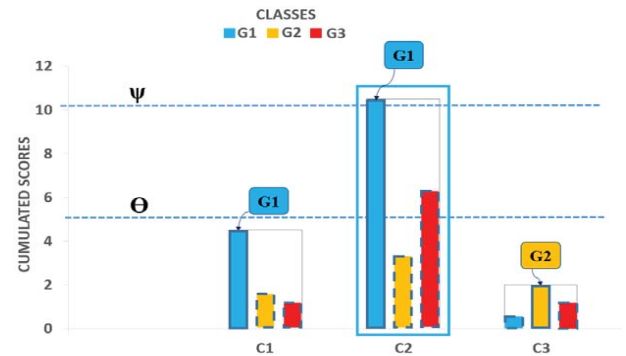
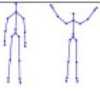
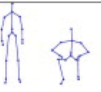
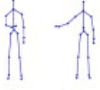



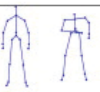


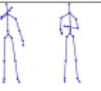
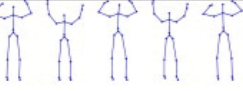



Illustration of the **global histogram** functioning at frame 7 with three classifiers which can G1, G2 or G3

© eric.anquetil@irisa.fr

## ■ DataSet: MSRC-12 dataset

- The Microsoft Research Cambridge-12 dataset (MSRC-12): sequences of skeleton data, represented as 20 joint locations.
  - *S. Fothergill, H. Mentis, P. Kohli, S. Nowozin, Instructing people for training gestural interactive systems, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, pp. 1737–1746.*
  - **12 gestures** performed by **30 subjects**
  - **594** sequences (about 50 sequences per class)
  - a single gesture is performed **several times along a sequence**.
- Participants were provided with 5 **instruction** modalities including:
  - images, text, video, images + text, and video + text.
- The dataset is annotated with **action points**
  - a pose within the gesture that clearly identifies its completion.

Metaphoric gestures	Main frames	Iconic gestures	Main frames
Start music\raise volume (G1)		Crouch or hide(G2)	
Navigate to next menu(G3)		Put on night vision goggles(G4)	
Wind up the music(G5)		Shoot with a pistol(G6)	
Take a bow to end the session(G7)		Throw an object such as a grenade(G8)	
Protest the music(G9)		Change weapon(G10)	
Lay down the tempo of a song(G11)		Kick to attack an enemy(G12)	

[Xi Chen, Markus Koskela 2015]

© eric.anquetil@irisa.fr

- Protocol (MSRC-12 dataset )
  - According to the leave-subjects-out protocol.
  - Mean  $F_{score}$  and its standard deviation is reported for each instruction modality.
- Other approaches
  - ELS = Efficient Linear Search;
  - RF = Random Forests;
  - RTMS = Real-Time Multi-Scale;
  - SSS =Structured Streaming Skeleton.
  - CuDi3D [Boulahia 2017]

$$F_{score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

	ELS [12]	RF [3]	RTMS [11]	SSS [4]	ELS [13]	<b>CuDi3D</b>
Video - Text	0.645 ± 0.149	0.679 ± 0.035	0.713 ± 0.105	0.707 ± 0.170	0.790 ± 0.133	<b>0.848 ± 0.060</b>
Images - Text	0.581 ± 0.134	0.563 ± 0.045	0.656 ± 0.122	0.730 ± 0.148	0.711 ± 0.228	<b>0.744 ± 0.072</b>
Text	0.437 ± 0.170	0.479 ± 0.104	0.521 ± 0.072	<b>0.713 ± 0.191</b>	0.622 ± 0.246	0.695 ± 0.080
Video	0.580 ± 0.189	0.627 ± 0.052	0.635 ± 0.075	0.557 ± 0.291	0.726 ± 0.225	<b>0.816 ± 0.060</b>
Images	0.497 ± 0.122	0.549 ± 0.102	0.596 ± 0.103	0.666 ± 0.194	0.670 ± 0.254	<b>0.719 ± 0.087</b>
Overall	0.548	0.579	0.624	0.675	0.704	<b>0.764</b>

© eric.anquetil@irisa.fr



## ■ Evaluation measure

Test Outcome		Desired Positive	Desired Negative
	Positive $N_E$	True Positive $N_E^A$	False Positive $N_R^A$
	Negative $N_R$	False negative $N_E^R$	True Negative $N_R^R$

### ■ Recognition/Error Rates

- TAR: True Acceptance Rate
- FAR: False Acceptance Rate

$$TAR = \frac{N_E^A}{N_E}$$

$$FAR = \frac{N_R^A}{N_R}$$

### ■ Accuracy Rates ("fiabilité")

- Global performance point of view

$$Accuracy = \frac{N_E^A + N_R^R}{N_E + N_R}$$

### ■ recall ("rappel")

- *information retrieval* → the number of relevant documents retrieved by a search / the total number of existing relevant documents

$$Recall = TAR$$

### ■ Precision ("précision")

- the number of items correctly labeled ∈ the positive class / the total number of elements labeled ∈ the positive class
- *information retrieval* → number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search

$$Precision = \frac{N_E^A}{N_E^A + N_R^A}$$

- The **F-Score** (or F Measure) conveys the balance between the precision and the recall.

$$F_{score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

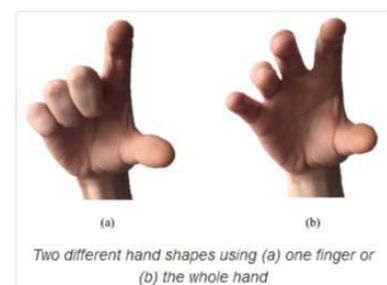
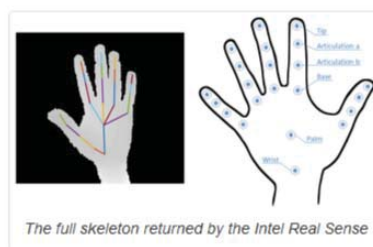
## ■ DHG DATASET: Dynamic Hand Gesture

### ■ DHG is a recent dynamic hand gesture dataset

- [De Smedt 2016] Quentin De Smedt, Hazem Wannous, and Jean-Philippe Vandeboor. Skeleton-based dynamic hand gesture recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 1–9, 2016.

### ■ 14 pre-segmented hand gestures

- performed in two ways: using **one finger** and **the whole hand**.
- Each gesture is performed between 1 and 10 times by **28 participants**
  - in 2 ways (one finger / the whole hand)
  - resulting in **2800 instances**.
- Each frame of sequences contains
  - a **depth image**
  - the **coordinates** of 22 joints both in the 2D depth image space and in the 3D world space forming a **full hand skeleton**.



- DHG: Segmented Gesture recognition in real-time streaming
  - COMPARISON BETWEEN
    - [Boulahia 2017] **HIF 3D APPROACH**
    - AND PREVIOUS APPROACHES
  - CONSIDERING 14 AND 28 GESTURES ON DHG\* DATASET

Method	14 gestures (%)	28 gestures (%)
HoWR [3]	35.61	-
SoCJ [3]	63.29	-
HoHD [3]	67.64	-
Oreifej and Liu [12, 14]	78.53	74.03
Devanne et al. [5, 14]	79.61	62.00
SoCJ + HoHD [3]	82.29	-
Guerry <i>et al.</i> [14]	82.90	71.90
SoCJ + HoHD + HoWR [3]	83.07	80.00
Ohn-Bar and Trivedi [11, 14]	83.85	76.53
De Smedt et al. [3, 14]	88.24	<b>81.90</b>
<b>Our</b>	<b>90.48</b>	80.48

[SHREC 2017] Results of the SHREC 2017 challenge on dynamic hand gesture recognition

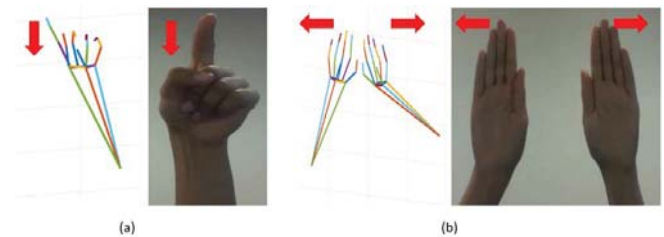
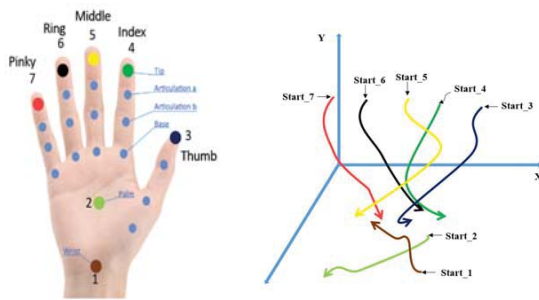
[Boulahia,IPTA 2017] Dynamic hand gesture recognition based on 3D pattern assembled trajectories. In 7th IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA 2017).

- CONFUSION MATRIX USING 14 GESTURES OF DHG DATASET

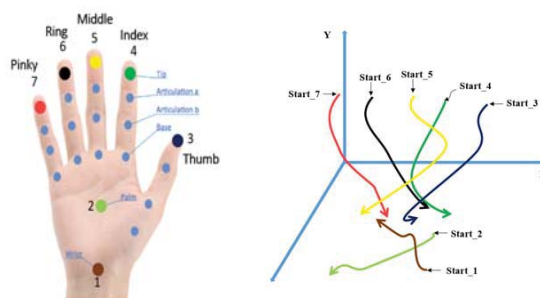
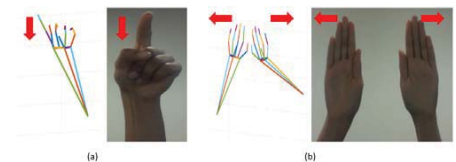
G	87.9	3.4	0.0	5.2	1.7	0.0	0.0	0.0	0.0	1.7	0.0	0.0	0.0	0.0
E	11.5	63.9	1.6	9.8	1.6	3.3	0.0	0.0	0.0	4.9	0.0	0.0	3.3	0.0
P	1.8	1.8	94.5	0.0	0.0	0.0	0.0	0.0	1.8	0.0	0.0	0.0	0.0	0.0
R-CW	13.7	2.0	0.0	82.4	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
R-CCW	1.8	1.8	0.0	0.0	89.1	1.8	5.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T	3.4	0.0	0.0	0.0	0.0	91.4	0.0	5.2	0.0	0.0	0.0	0.0	0.0	0.0
S-R	0.0	0.0	0.0	0.0	1.6	0.0	98.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
S-L	0.0	0.0	0.0	1.9	3.7	0.0	0.0	94.4	0.0	0.0	0.0	0.0	0.0	0.0
S-U	0.0	1.5	11.8	0.0	0.0	1.5	0.0	0.0	83.8	1.5	0.0	0.0	0.0	0.0
S-D	0.0	1.6	0.0	9.8	0.0	0.0	0.0	0.0	0.0	86.9	0.0	0.0	1.6	0.0
S-X	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.4	0.0	0.0	98.6	0.0	0.0	0.0
S-V	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.8	98.2	0.0	0.0
S-+	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.7	0.0	0.0	98.3	0.0
Sh	0.0	0.0	2.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	97.3
	G	E	P	R-CW	R-CCW	T	S-R	S-L	S-U	S-D	S-X	S-V	S-+	Sh

- Weaknesses of existing dynamic hand gesture datasets:
  - Composed of very short clips (around 30 frames)
  - Gestures are performed with a single hand
  - Perfectly denoised, with almost no missing motion segments
  - Composed of pre-segmented gestures only

- LMDHG dataset:
  - A leapMotion (NON-) Segmented DataSet



- LMDHG dataset: A leapMotion DataSet
  - Composed of 50 unsegmented sequences of gestures performed with either one hand or both hands by 21 participants
  - Each sequence contains  $13 \pm 1$  class gestures leading to a total of 608 gesture instances
  - Order of class in each sequence is aleatory
  - Each frame contains the 3D coordinates of 46 joints
  - Ground truth Start/End along with the class labels are provided
  - LMDHG dataset contains noisy and incomplete gestures.



Gesture	#Hands	tag name
Point to	1	HG1
Catch	1	HG2
Shake with two hands	2	HG3
Catch with two hands	2	HG4
Shake down	1	HG5
Shake	1	HG6
Draw C	1	HG7
Point to with two hands	2	HG8
Zoom	2	HG9
Scroll	1	HG10
Draw Line	1	HG11
Slice	1	HG12
Rotate	1	HG13

## ■ CONFUSION MATRIX ON THE COLLECTED **LMDHG** DATASET

- [Boulahia,IPTA 2017] Dynamic hand gesture recognition based on 3D pattern assembled trajectories. In 7th IEEE International Conference on Image Processing Theory, Tools and Applications (IPTA 2017).

- **Protocol:** train on 70% of the sequences,
  - Train i.e. sequences from 1 to 35
  - Test on the remaining 15 sequences.
- **Overall score:**
  - **Segmented** : 84.78%

HG1	92.9	7.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HG2	6.7	80.0	0.0	6.7	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
HG3	0.0	0.0	92.9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.1
HG4	0.0	6.7	0.0	86.7	0.0	0.0	0.0	0.0	6.7	0.0	0.0	0.0	0.0
HG5	0.0	6.7	0.0	0.0	66.7	0.0	0.0	0.0	0.0	0.0	0.0	20.0	6.7
HG6	0.0	0.0	0.0	0.0	0.0	85.7	0.0	0.0	0.0	0.0	7.1	0.0	7.1
HG7	0.0	0.0	6.7	0.0	0.0	0.0	93.3	0.0	0.0	0.0	0.0	0.0	0.0
HG8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
HG9	0.0	0.0	0.0	0.0	0.0	0.0	8.3	0.0	83.3	0.0	0.0	0.0	8.3
HG10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	92.9	0.0	0.0	7.1
HG11	0.0	6.7	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	86.7	0.0	0.0
HG12	0.0	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	6.7	0.0	86.7	0.0
HG13	0.0	0.0	6.7	0.0	6.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	86.7
	HG1	HG2	HG3	HG4	HG5	HG6	HG7	HG8	HG9	HG10	HG11	HG12	HG13

## ■ Experimental results on LMDHG dataset : Unsegmented gestures

- BaseLine with a basic approach
  - A sliding window approach in which the window size equals to the average of training instances
- Protocol
  - train on 70% of the sequences, i.e. sequences from 1 to 35
  - test on the remaining 15 sequences.
- For evaluating this basic approach with unsegmented sequences, we use the Fscore :
  - **Overall Fscore:** 54.11%

$$F_{score} = 2 * \frac{Precision * Recall}{Precision + Recall}$$

## \_Chapitre 12

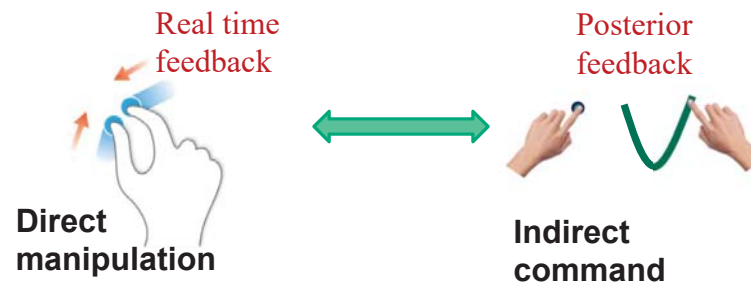
# Early Recognition

121

### \_Chap. 12 | 2D and 3D Action/Gesture recognition: a challenge ?

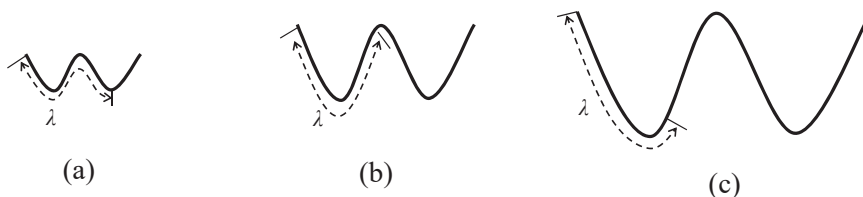
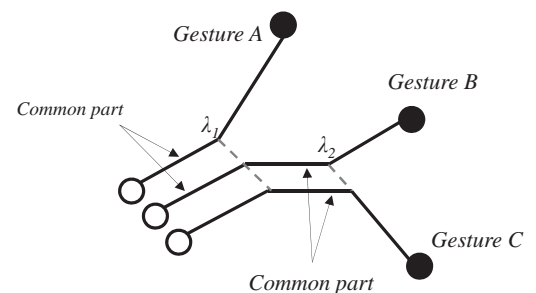
- ❖ **Introduction: understand the problematic of gesture interaction**
  - What is a gesture: the different natures of gestures
  - Human Computer Interaction: new opportunities
- ❖ **Gesture recognition: Isolated Gestures Classification (segmented)**
  - Overview of the task: recognizing isolated gestures (The overall pattern recognition process)
  - Machine Learning and Pattern recognition: a short overview of some existing techniques
    - Gesture classification: "Time-series" approaches
    - Pre-segmented Action Recognition: Skeleton based and "Statistical" approaches
- ❖ **Gesture recognition in real-time streaming (non segmented)**
  - Overview of the task: recognizing in real-time streaming
  - Non-segmented Action Recognition: Example of one approche [Boulahia 2017]
  - Presentation of experimental results using Kinect and Leap Motion
- ❖ **Early Gesture recognition**

- One possible Goal for Early recognition:
  - To merge **Direct** and **Indirect** interactions into a same interface
  - we have to distinguish gesture in the very beginning part

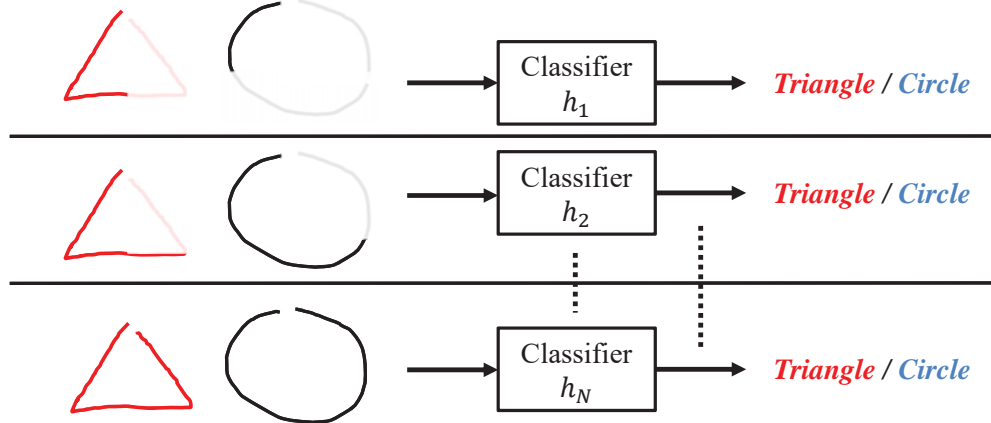


- One Solution:
  - a **reject option based multi-classifier** system
  - for handwritten gesture early recognition [Zhaoxin 2016]

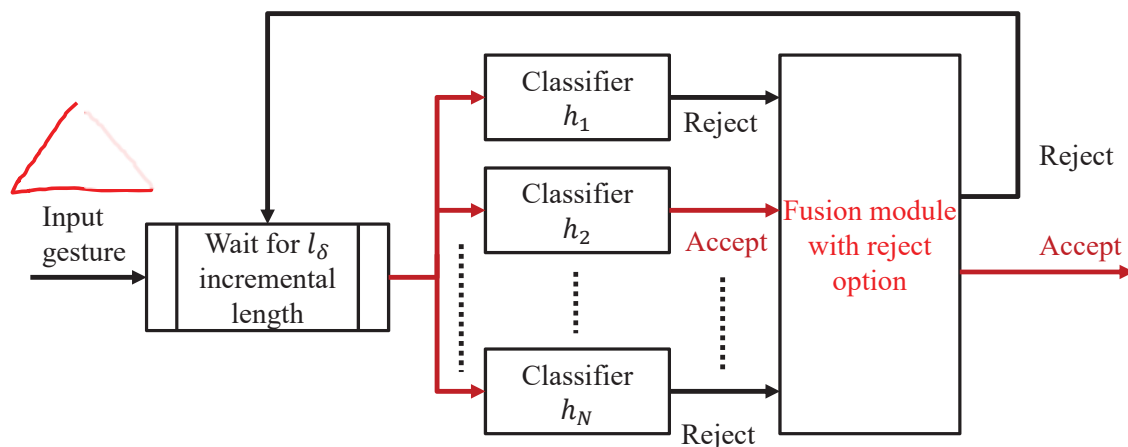
- **Goal:** recognize the gesture
  - from their early part
  - instead of waiting until the end of them.
- **Difficulties**
  - to deal with the **common beginning part ambiguity**
  - The proportion of the earliness is unpredictable
    - (a) A normalized gesture as a template.
    - (b) (c) In a size free context.



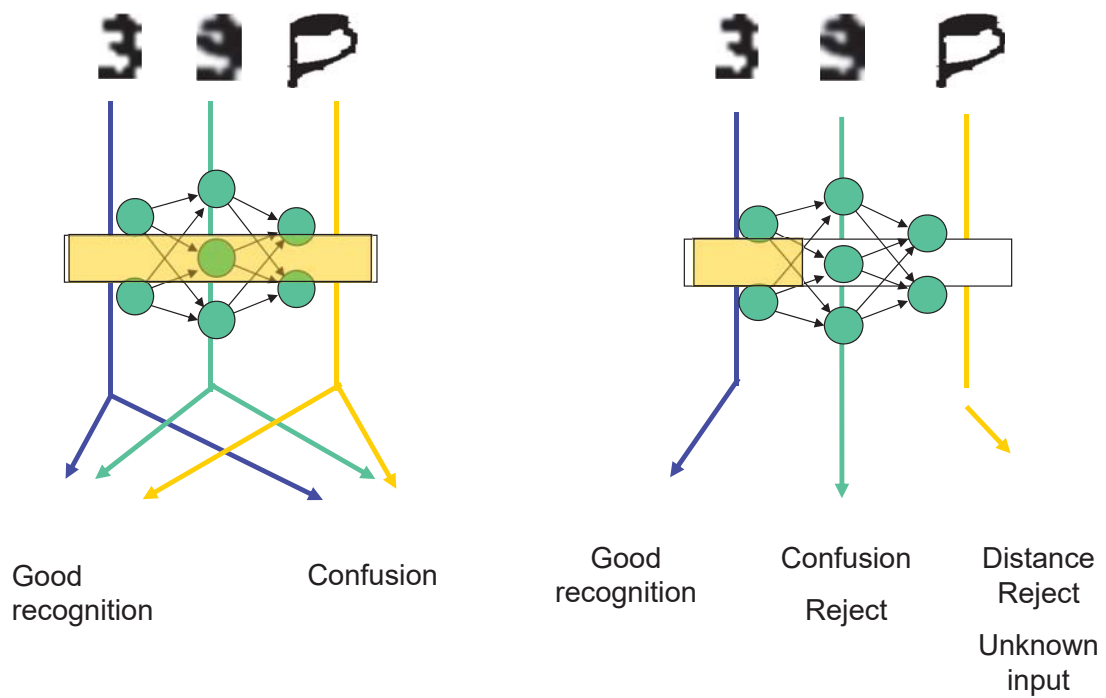
- During the training, each classifier dedicates to different part of gestures (short, medium, long)



- **One strategy: A reject option based multi-classifier early recognition system**
  - All classifiers try to recognize the gestures
  - The fusion module merge trustable decisions
- Two types of reject are used to evaluate the confidence
  - - **ambiguity**: the shape looks like beginning of several different gesture classes
  - - **outlier**: the classifier has never seen this type of shape





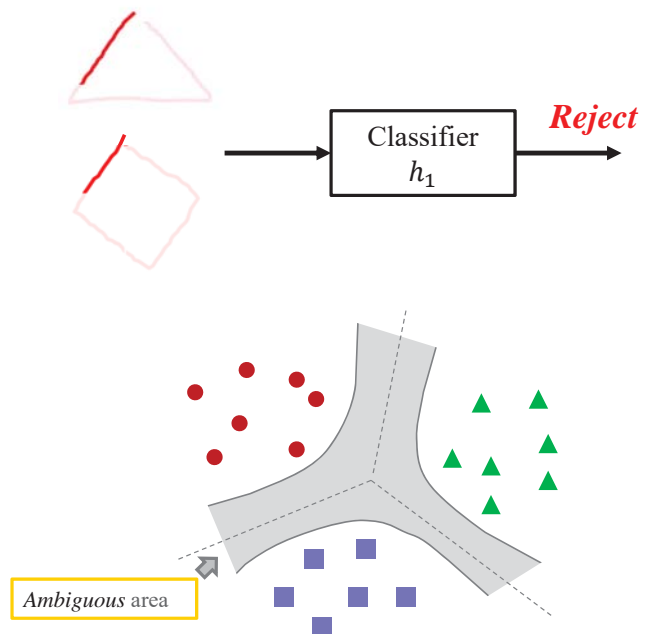


■ Ambiguity rejection [5]

$$\psi_i^{Amb} = \frac{p_i - p_j}{p_i}$$

where  $p_i$  is the confidence value of best class,  
 $p_j$  is the second best class from the classifier.

[5] H. Mouchère and E. Anquetil. **A unified strategy to deal with different natures of reject.** In *Pattern Recognition*, 2006. ICPR 2006, volume 2, pages 792-795, 2006.





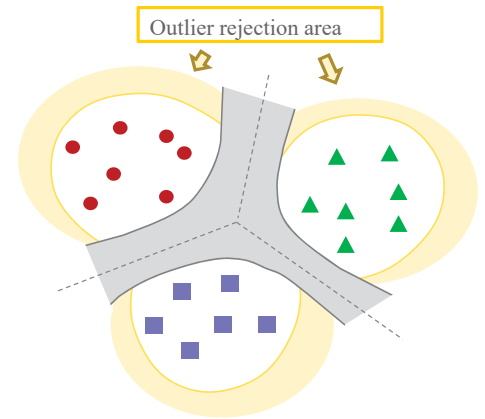
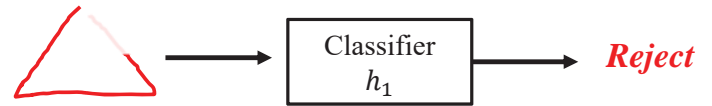
## ■ Outlier rejection

Estimate the outlier confidence value

using the minimum distance to the prototypes:

$$D_i = \min_{j \in N} (d(g_t, g_i^j))$$

$g_t$  is a test sample,  $g_i$  is the prototype sample of class  $i$ ,  $N$  is the number of prototypes.



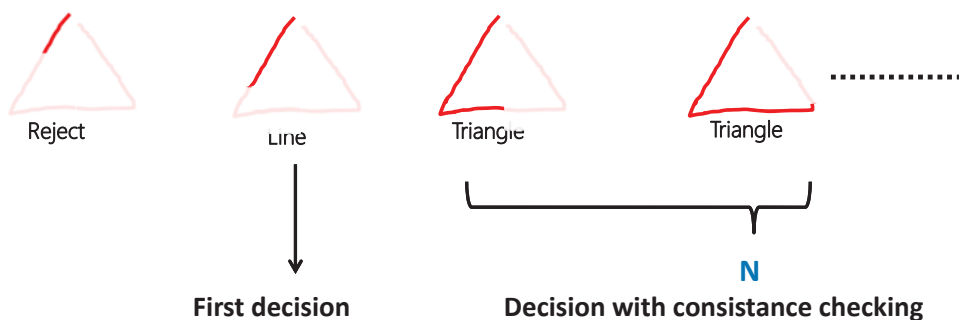
## ■ Reliability function

$$\psi_i^{out} = \begin{cases} e^{-\frac{(D_i - \mu)^2}{2\sigma^2}} & \text{if } D_i \geq \mu \\ 1 & \text{if } D_i < \mu \end{cases}$$

Where  $\mu$  and  $\sigma$  is the minimum distance and deviation computed from validation set.

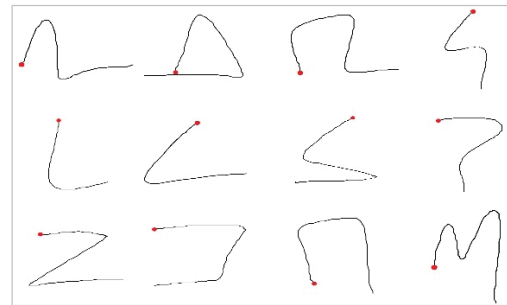
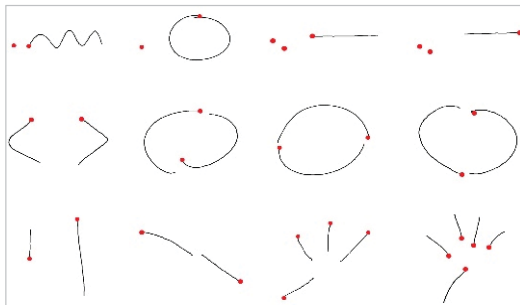
## ■ Dynamic decision with consistence checking (N)

- N consecutive identical results in the stream of outputs
- Several recognitions during the drawing with more and more information



## ■ Examples of Gestures: MGSet/ILG datasets

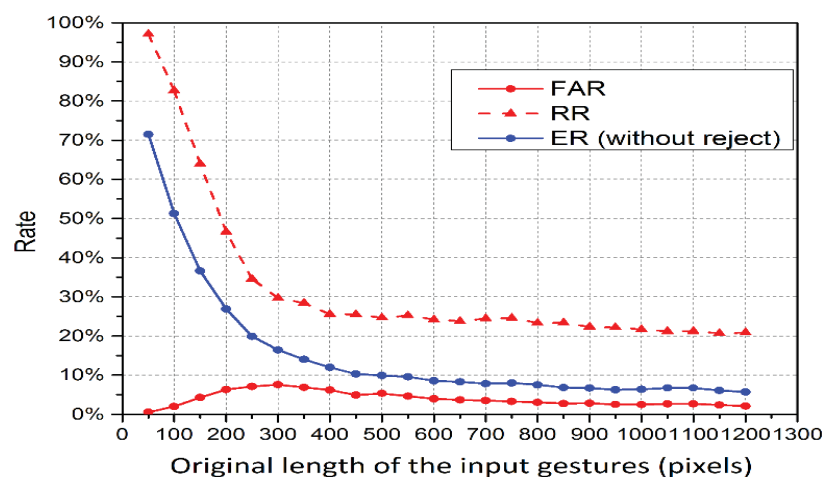
- (MGSet) Multi-stroke gestures (45 classes, 33 users, 6K samples)
- (ILG) Single-stroke gestures (45 classes, 21 users, 2K samples)



## ■ Results (MGSet)

- (MGSet) Multi-stroke gestures (45 classes, 33 users, 6K samples)
- Results with decision consistence:  
**reject opt. allows to improve earliness**

N	With Reject Option (MGSet)				
	TAR	FAR	RR	Earliness	Avg. T (ms)
1	81.89%	14.56%	3.54%	37.04%	456.21
2	83.44%	10.85%	5.71%	46.82%	523.34
3	82.38%	8.85%	8.77%	55.89%	591.33



## \_Chapitre 13

# Fuzzy Clustering

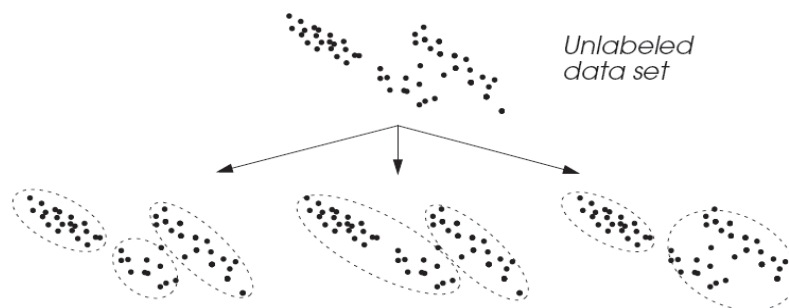
133

### \_Chap. 13 | Fuzzy clustering: Introduction

134

#### ○ What is cluster analysis ?

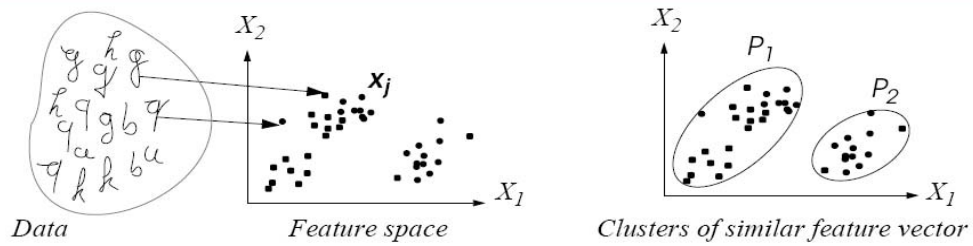
*“partitioning a collection of data points into a number of subgroups (clusters), where the objects inside a cluster show a **certain degree of closeness or similarity**”*



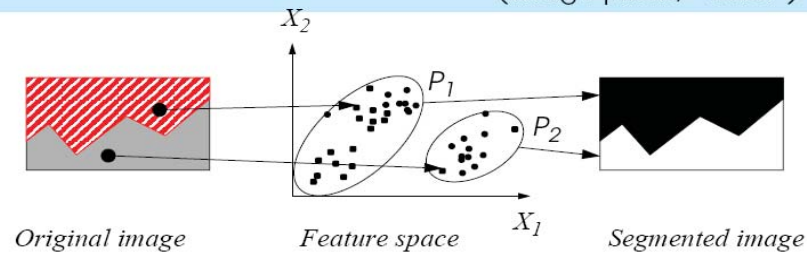
#### ⇒ Major difficulties to find “natural groupings”:

- ✓ Large variability in cluster shapes
  - Classification criterion - Similarity or distance measure
- ✓ Number of clusters ?
  - Cluster validity problem

○ Pattern Recognition (patterns, (curvature, relative dimension, ...)) → classes



○ Image Segmentation (image pixels, "color") → regions



○ Medical Diagnosis (patients, symptoms) → diseases

## 1. Data Representation and Notation

- Features
- Partitions

## 2. Clustering Methods

- Different clustering families
- Principles of alternating optimization
  - Hard C-Means
  - Fuzzy C-Means
  - Possibilistic Clustering
- Cluster Validity

## 3. Discussion and Application

### ○ Notation

- let  $D = \{x_j / j=1 \dots N\}$  be the data set of  $N$  items  $x_j$
- let  $P = \{P_i / i=1 \dots C\}$  be the  $C$  cluster prototypes
- Each  $x_j$  is described by a feature vector:  $x_j = (x_{j1}, x_{j2}, \dots, x_{jn})^T$

(Data, Feature space)  $\rightarrow$  clusters  
 $(x_j, x_j = (x_{j1}, x_{j2}, \dots, x_{jn})^T) \rightarrow P_i$

### ○ C-Partition

- A **C-partition** can be represented by a  $(C \times N)$  matrix  $U = (\mu_{ij})$ , where  $\mu_{ij}$  represents membership of  $x_j$  in  $P_i$

$$U = \begin{matrix} & \text{data points} \\ \text{Clusters} & \begin{bmatrix} \mu_{11} & \dots & \mu_{1j} & \dots & \mu_{1N} \\ \dots & \dots & \dots & \dots & \dots \\ \mu_{i1} & \dots & \mu_{ij} & \dots & \mu_{iN} \\ \dots & \dots & \dots & \dots & \dots \\ \mu_{C1} & \dots & \mu_{Cj} & \dots & \mu_{CN} \end{bmatrix} \end{matrix}$$

- A clustering algorithm  $\equiv$  finds the  $\{U_{HCM}, U_{FCM}, U_{PCM}\}$  which "best" explains and represent the structure in  $X$ .

### ○ Different partition properties

#### ⇒ constrained crisp partition:

$$U_{HCM} \equiv \mu_{ij} \in \{0,1\}, \quad 0 < \sum_{j=1}^N \mu_{ij} < N, \quad \sum_{i=1}^C \mu_{ij} = 1$$

#### ⇒ constrained fuzzy partition:

$$U_{FCM} \equiv \mu_{ij} \in [0,1], \quad 0 < \sum_{j=1}^N \mu_{ij} < N, \quad \sum_{i=1}^C \mu_{ij} = 1$$

#### ⇒ unconstrained fuzzy partition:

$$U_{PCM} \equiv \sum_{i=1}^C \mu_{ij} = 1$$

*do not necessarily sum up to one over any column*

$$U_{HCM} \subset U_{FCM} \subset U_{PCM}$$

○ Probabilistic Clustering

*Example: Gaussian mixture decomposition*

○ Competitive Learning

Neural network based algorithms

*Example: Self Organization Map (SOM)*

○ Vector Quantization

*Example: LBG algorithm*

○ Alternating optimization

Clustering methods based on objective function

*Example: Fuzzy C-Means algorithm*

>> Many common points between these different approaches <<

○ General principle

*“Alternating clustering methods are based on an iterative minimization of a criterion function (objective function) to extract a partition of the data set”*

○ General iterative algorithm / alternating optimization

✓ step 1 (Initialization)

- Fix  $C$ , initial  $C$ -partition, ...

✓ step 2 (Prototype adaptation)

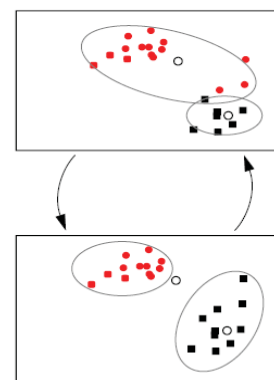
- Calculate the  $C$  prototypes  $P_i$

✓ step 3 (Update the  $C$ -partition)

- “Label” evaluation of the data
- Update the  $C$ -partition matrix  $U$

✓ step 4 (Termination)

- Repeat steps 2-4 until the termination criterion is met



⇒ Based on a **constrained crisp partition**:

$$\mu_{ij} \in \{0,1\}, \quad 0 < \sum_{j=1}^N \mu_{ij} < N, \quad \sum_{i=1}^C \mu_{ij} = 1$$

⇒ The Objective function is the WGSS:

$$J_{P,U,D} = \sum_{i=1}^C \sum_{x_j \in P_i} d^2(x_j, P_i)$$

○ HCM algorithm (Duda and Hart (Dud73))

- ✓ **step 1 (Initialization)**
  - Fix  $2 \leq C < N$ , initial **C-partition**  $U^{(0)}$
- ✓ **step 2 (Prototype adaptation)**
  - Calculate the C prototypes  $P_i$
- ✓ **step 3 (Update the C-partition)**
  - Update the C-partition matrix  $U^{(t)}$
- ✓ **step 4 (Termination)**
  - Repeat steps 2-4 until  $\Delta U < \varepsilon$

$$P_i = \frac{\sum_{j=1}^N \mu_{ij} x_j}{\sum_{j=1}^N \mu_{ij}}$$

$$\mu_{ij}^{(t+1)} = \begin{cases} 1, & d(x_j, P_i^{(t)}) = \min_{1 \leq k \leq C} (d(x_j, P_k^{(t)})) \\ 0, & \text{otherwise} \end{cases}$$

⇒  $\mu_{ij} \in \{0,1\}, \quad \sum_{i=1}^C \mu_{ij} = 1$  : means that each  $x_j$  is in exactly one of the C clusters.

⇒  $0 < \sum_{j=1}^N \mu_{ij} < N$  : means that no cluster is empty and no cluster is all of X.

⇒ The objective function is the classical WGSS (Within Group Sum of Squared errors)

$$J_{P,U,D} = \sum_{i=1}^C \sum_{j=1}^N \mu_{ij} d^2(x_j, P_i)$$

⇒ where  $d^2$  represents a distance measure, for example the euclidean distance measure:

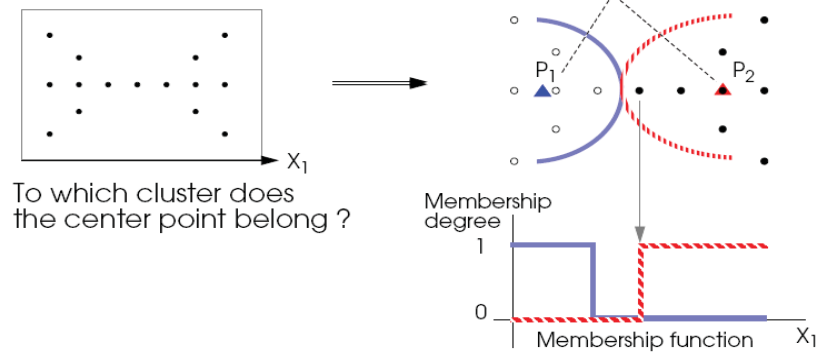
$$d^2(x_j, P_i) = \|x_j - P_i\|^2 = \sum_{k=1}^n (x_{jk} - P_{ik})^2$$

⇒ The second version is based on:  
find the centroid → reallocate the cluster memberships to minimize the errors between the data and the prototypes.

$$P^{(t-1)} \rightarrow U^{(t)} \rightarrow P^{(t)}$$



### ○ Classical example (The butterfly)



### ○ Discussion

- ⇒ In HCM clustering every point belongs to only 1 cluster (**constrained crisp partition**)
- ⇒ Transition between full membership and no membership is abrupt
- ⇒ Hard decisions on class assignments
- ⇒ Consequently the 2 clusters can not be symmetric with respect to the center point

- ⇒ Based on a **constrained fuzzy partition**:

$$\mu_{ij} \in [0,1], \quad 0 < \sum_{j=1}^N \mu_{ij} < N, \quad \sum_{i=1}^C \mu_{ij} = 1$$

- ⇒ The Objective function is

$$J_{P,U,D} = \sum_{i=1}^C \sum_{j=1}^N \mu_{ij}^m d^2(x_j, P_i)$$

### ○ FCM algorithm (Bezdek (Bez81))

- ✓ **step 1 (Initialization)**
  - Fix  $2 \leq C < N$ ,  $1 \leq m < \infty$ , initialize  $U(0)$
- ✓ **step 2 (Prototype adaptation)**
  - Calculate the  $C$  prototypes  $P_i$
- ✓ **step 3 (Update the C-partition)**
  - Update the C-partition matrix  $U(t)$
- ✓ **step 4 (Termination)**
  - Repeat steps 2-4 until  $\Delta U < \epsilon$

$$P_i = \frac{\sum_{j=1}^N (\mu_{ij})^m x_j}{\sum_{j=1}^N (\mu_{ij})^m}$$

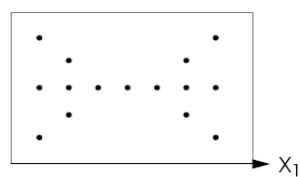
$$\mu_{ij} = \frac{1}{\sum_{k=1}^C \left( \frac{d^2(x_j, P_i)}{d^2(x_j, P_k)} \right)^{\frac{1}{m-1}}}$$

- ⇒ transition between full membership and no membership is gradual rather than abrupt.
- ⇒  $\mu_{ij}$  represent membership degrees
- ⇒ soft decisions on class assignments

#### ○ Parameters of Fuzzy C-means:

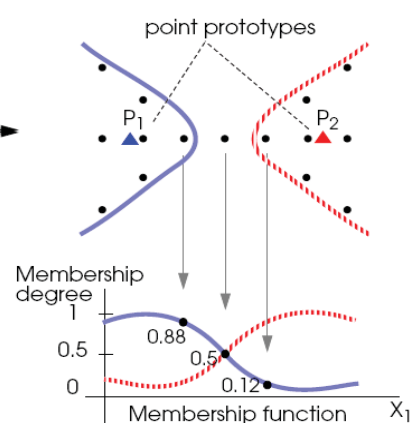
- ⇒ **C** : Number of clusters
- ⇒  $U(0)$  : Initial C-partition
- ⇒  $d^2(x_j, P_i) = (x_j - P_i)^T A (x_j - P_i)$  : "distance measure"
  - If  $A = \text{Identity matrix}$  then  $d^2$  is the Euclidean Norm
- ⇒ **m** is the weighting exponent called the "fuzzifier"
  - When  $m \rightarrow 1$ , Fuzzy C-Means solution become hard.
  - to control the "fuzziness" of the resulting clusters

#### ○ The butterfly example



Parameters :

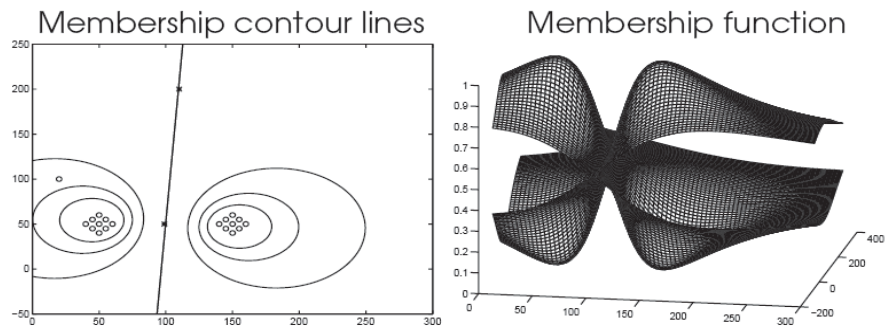
- $m = 2.0$
- Euclidean Norm



#### ○ Discussion

- ⇒ In FCM clustering every point belongs to every cluster to different degrees ( $\Leftrightarrow$  **constrained fuzzy partition**)
- ⇒ Minimization of the error propagation during the iterative optimization ( $\Leftrightarrow$  **soft decision in each iteration**)

### ○ Example

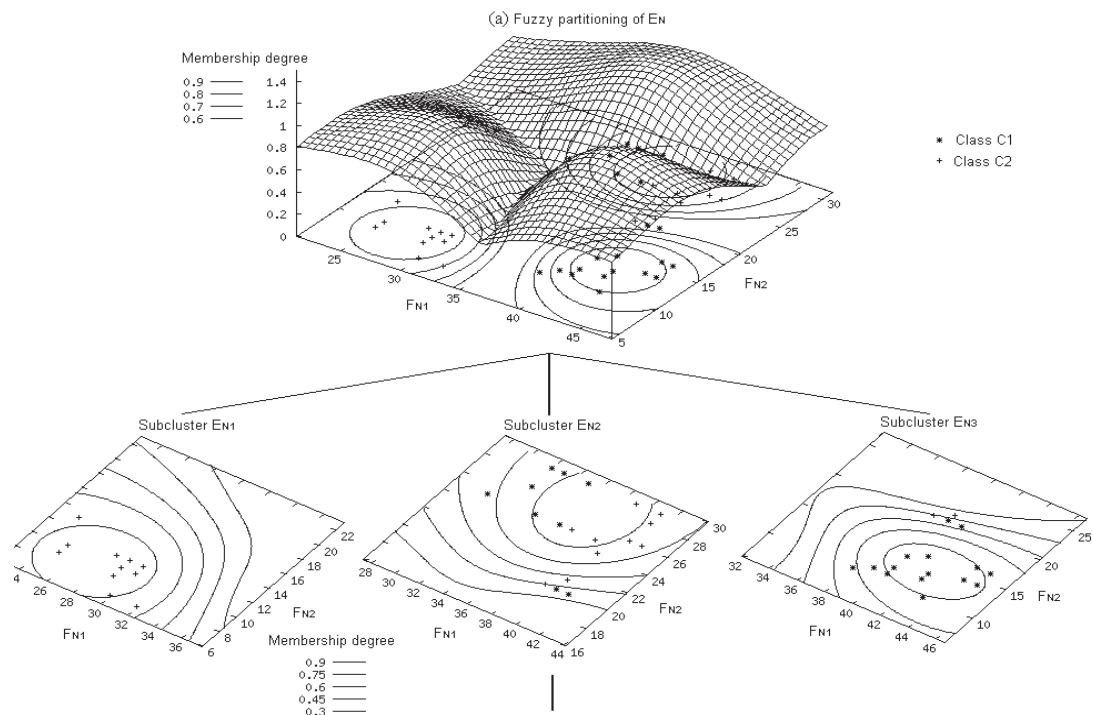


### ○ Interpretation

- ⇒ Memberships can be interpreted as between class degrees of **sharing**
- ⇒ The centers (prototypes) do not coincide with the true centers of the clusters
- ⇒ Influence of noise points

### ○ Useful for

*the discrimination of clusters → extraction and modeling of the best boundaries between clusters*



⇒ Based on a **unconstrained fuzzy partition**:

$$\sum_{i=1}^C \mu_{ij} = 1$$

⇒ The Objective Function is

$$J_{P, U, X} = \sum_{i=1}^C \sum_{j=1}^N \mu_{ij}^m d^2(x_j, P_i) + \sum_{i=1}^C \eta_i \sum_{j=1}^N (1 - \mu_{ij})^m$$

○ Algorithm (Krishnapuram&Keller (Kri94a) (Kri93b))

✓ **step 1 (Initialization)**

- Fix  $2 \leq C < N$ ,  $1 \leq m < \infty$ , initialize  $U(0)$

✓ **step 2 (Prototype adaptation)**

- Calculate the  $C$  prototypes  $P_i$

✓ **step 3 (Update the C-partition)**

- Update the C-partition matrix  $U(t)$

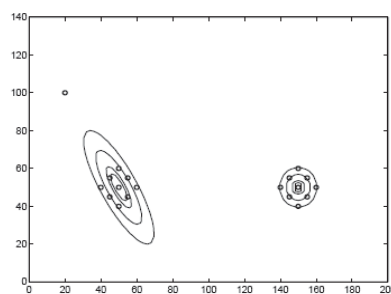
✓ **step 4 (Termination)**

- Repeat steps 2-4 until  $\Delta U < \epsilon$

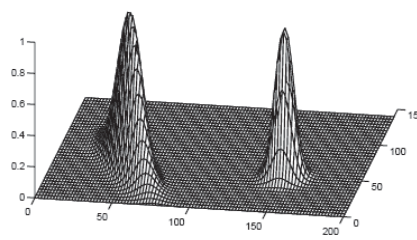
$$P_i = \frac{\sum_{j=1}^N (\mu_{ij})^m x_j}{\sum_{j=1}^N (\mu_{ij})^m}$$

$$\mu_{ij} = \frac{1}{1 + \left( \frac{d^2(x_j, P_i)}{\eta_i} \right)^{\frac{1}{m-1}}}$$

○ Example



Membership contour lines



Membership function

○ Interpretation

- ⇒ Memberships can be interpreted as degrees of **typicality** (absolute numbers)
- ⇒ The centers (prototypes) coincide with the "true" centers of the clusters
- ⇒ Low influence of Noise points

○ Useful for

*the intrinsic characterization of each clusters*

*“the determination of the optimal number of clusters present in the data is a difficult problem”*

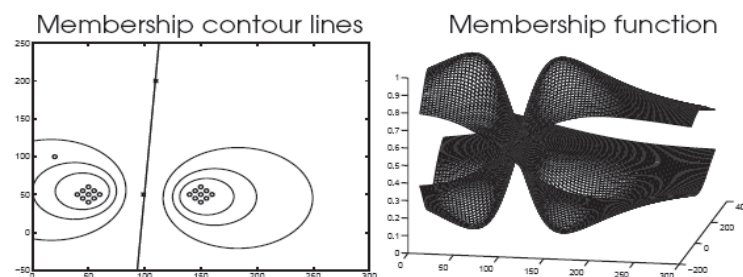
### ○ Cluster validity criterion

- ⇒ Many different criterions of cluster validity :
  - often based on a measure of **compactness** and **separability** of the clusters.

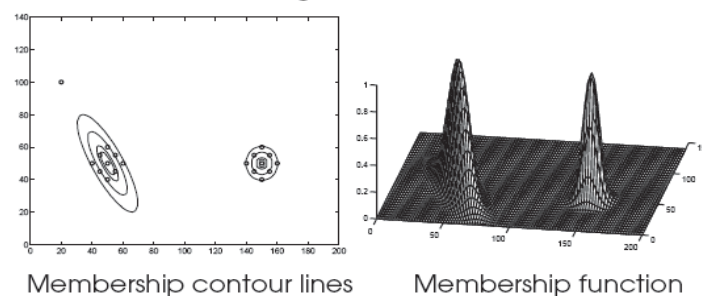
### ○ Different approaches

- ⇒ Iterative clustering:
  - try successively different values of **C** and evaluate the validity
- ⇒ Progressive clustering:
  - start with one cluster and try progressively to extract a new cluster
- ⇒ Agglomerative clustering:
  - start with many clusters and agglomerate the nearest clusters according to a neighborhood criterion.

### ○ Fuzzy C-means

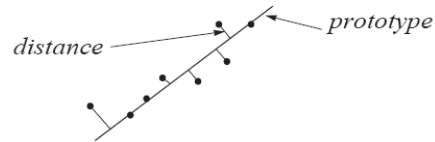


### ○ Possibilistic clustering



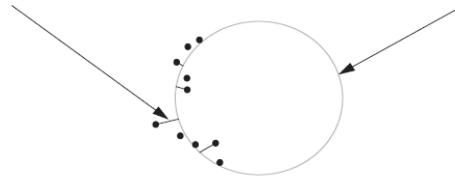
*“The extraction of shell-like clusters (with no interior points) needs the redesigning of the **distance measure** and/or of the **prototype** of each cluster.”*

- Example: The fitting of linear structure (e.g. lines)

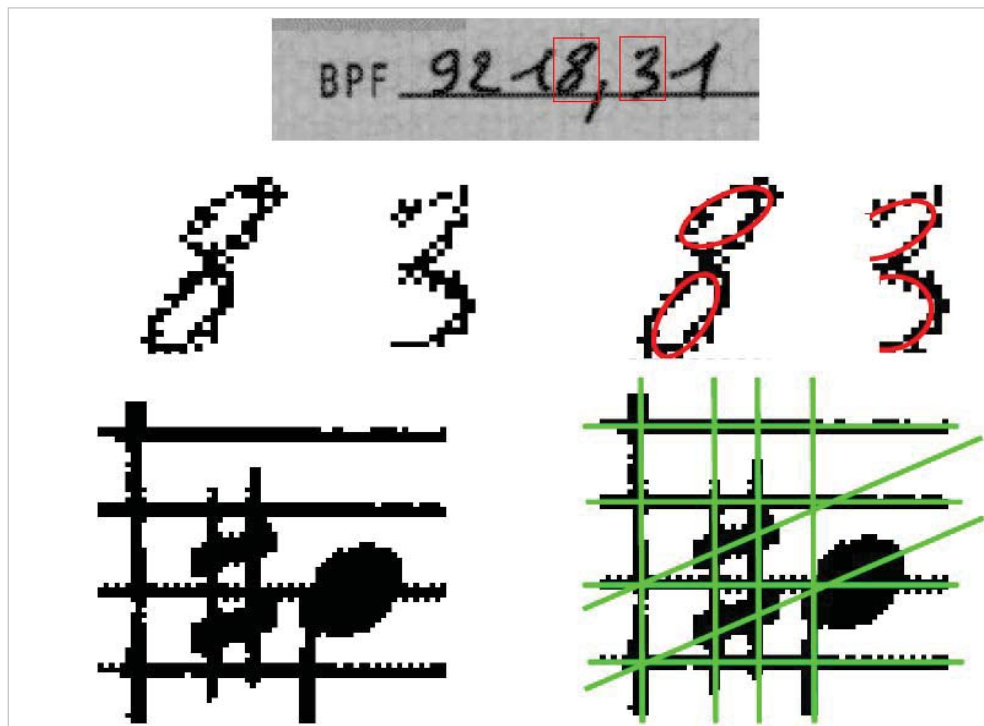


- Example: The fitting of circular shell

$$\text{distance : } d^2(x_j, c_i) = (\|x_j - c_i\| - r_i)^2 \quad \text{prototype : } (c_i, r_i)$$



⇒ Useful for the detection of boundaries and shapes of objects from images



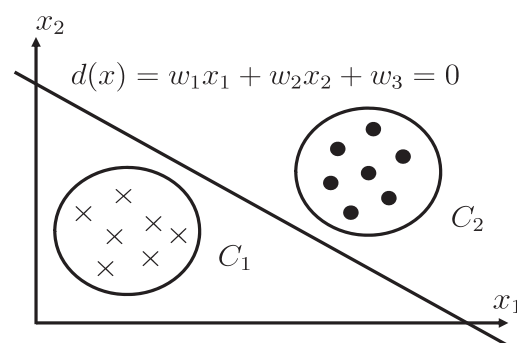
## \_Chapitre 15

# Classification: Linear Discriminant Functions

### \_Chap. 15 | Classification with Linear Discriminant Functions

156

- Pattern  $x = (x_1, x_2, \dots, x_n) \Rightarrow$  point in the n-dimensional vector space
- i.e. numerical features
- Assumption:  $x_i, 1 \leq i \leq n$ 
  - Classes take separable regions which can be separated by linear discriminant functions
  - Parametric models





- How does it work?
  - Labeled training data
  - Calculate discriminant function (e.g., perceptron algorithm)

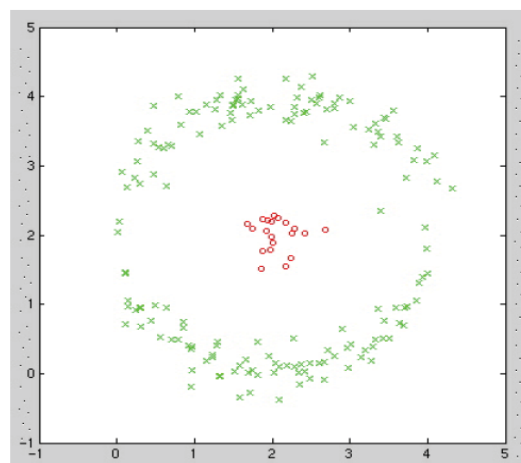
- Discriminant function

$$d(x) = w_1x_1 + \cdots + w_nx_n + w_{n+1} = wx^t = 0$$

- For an unknown pattern :  $x$

$$x \in \begin{cases} C_1, & \text{if } d(x) > 0 \\ C_2, & \text{if } d(x) < 0 \\ \text{reject,} & \text{if } d(x) = 0 \end{cases}$$

- Linear discriminant functions are not always sufficient
  - i.e. non linear hyperplanes are needed in  $\mathbb{R}^n$



■ Linear discriminant function

$$d(x) = w_1x_1 + \dots + w_nx_n + w_{n+1}$$

■ Generalized discriminant function

$$\begin{aligned} d(x) &= w_1f_1(x) + \dots + w_mf_m(x) + w_{m+1} \\ &= w(x^*)^t \end{aligned}$$

■ With

$$w = (w_1, \dots, w_m, w_{m+1})$$

$$x^* = (f_1(x), \dots, f_m(x), 1); \quad f_1(x), \dots, f_m(x) : \text{functions}$$

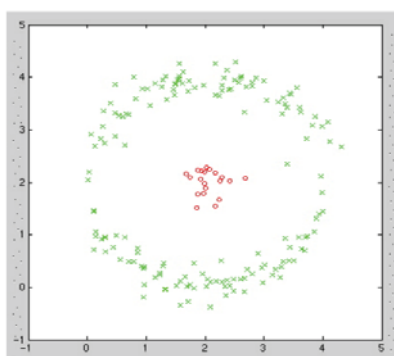
■ Procedure

- Reduce any arbitrary discriminant function of the above mentioned form to the linear form by transforming the given pattern by application of functions into  $f(x)$ .
- In general  $x^*$ , i.e. to enable linear separability transform patterns into a space of higher dimension.  
 $m \gg n$

■ Example of function

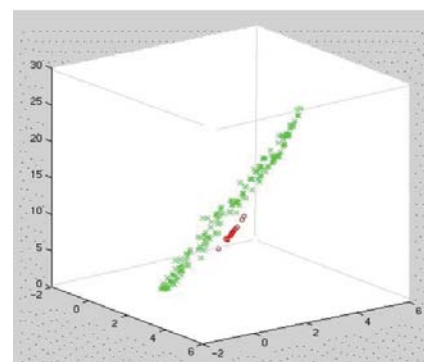
$$f(x)$$

[Thierry Artières]



2 dimensions

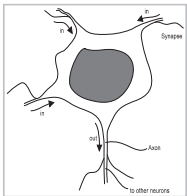
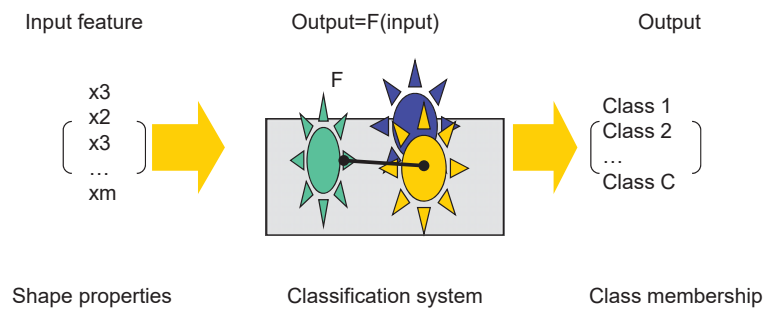
$$(x, y) \rightarrow (x, y, x^2 + y^2)$$

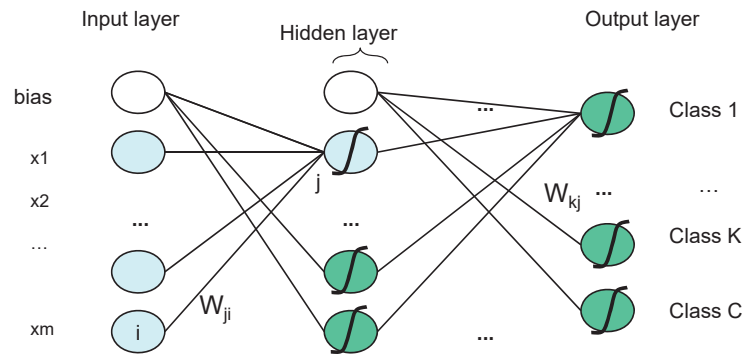


3 dimensions

\_Chapitre 16

Neural Networks

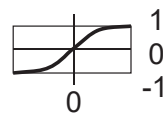




Input of a neural  $j$   
of the layer 0

$$a_j = \sum_{i=1}^m w_{ji} x_i + w_{j0}$$

$f$ : activation function  
(example sigmoid):



Output of neural  $j$   
 $y_j = f(a_j)$

Input of neural  $K$

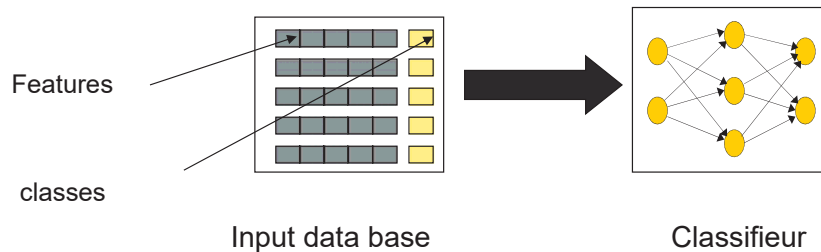
$$a_k = \sum_{j=1}^n w_{kj} y_j + w_{k0}$$

Output of neural  $k$   
 $z_k = f(a_k)$

## ■ Learning and generalization capacities

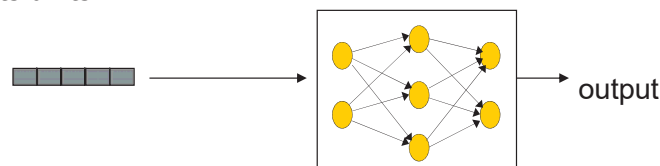
### ■ Learning

- consists of presenting an input pattern and modifying the network parameters (weights) to reduce distances between the computed output and the desired output



### ■ Generalization / Feedforward

- consists of presenting a pattern to the input units and passing the signals through the network in order to get outputs units

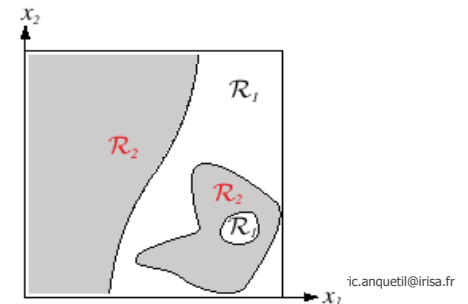
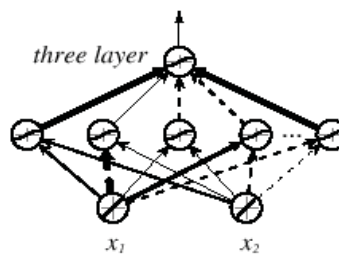
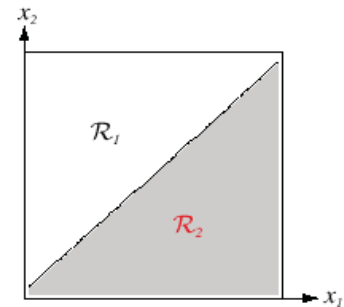
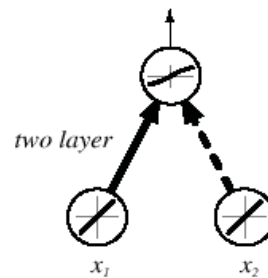


■ MLP: Universal approximator: [A. Kolmogorov]

- "Any continuous function from input to output can be implemented in a three-layer net, given sufficient number of hidden units, proper nonlinearities, and weights."

→ Any function from input to output can be implemented as a three-layer neural network

[Duda, PHart, Stork, "Pattern Classification"]



■ The aim

- Construction of a network :
  - to define the nonlinear functions and the weight values

■ The Learning process (supervised)

- Some empirical choices
  - Number of neural and layers
  - Activation functions
- Principles
  - Present the network a number of inputs and their corresponding outputs
  - See how closely the actual outputs match the desired ones
  - Modify the parameters to better approximate the desired outputs

## ■ Principle

- The error signal is obtained from the comparison between the target and estimated signal.
- The error signal is propagated layer by layer from the output layer to the input layer to adaptively adjust all weights in the MLP.

## ■ Back-propagation (BP) algorithm

- Let  $t_k$  be the k-th target (or desired) output and  $y_k$  be the k-th computed output with  $k = 1, \dots, c$  and  $w$  represents all the weights of the network

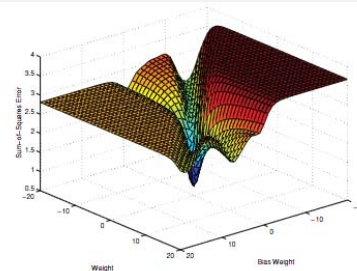
- The training error to minimize:

- Goal:

We go through the weight space to find the point corresponding to the minimum of the error

- Method: gradient descent

$$E(w) = \frac{1}{2} \sum_{k=1}^c (y_k - t_k)^2 = \frac{1}{2} \|y - t\|^2$$



© eric.anquetil@irisa.fr

- The backpropagation learning rule is based on gradient descent

$$\frac{\partial E}{\partial w} = \nabla E[\mathbf{w}] \equiv \left[ \frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_p} \right]$$

- Going back from "output" to "input":
  - 1 Calculate the derivatives of the error with respect to weights
  - 2 Using these derivatives for adjust the weights

$$\mathbf{w}^{(\tau+1)} \leftarrow \mathbf{w}^{(\tau)} - \eta \nabla E[\mathbf{w}^{(\tau)}]$$

$$\Delta w = -\eta \frac{\partial E}{\partial w}$$

where  $\eta$  is the learning rate which indicates the relative size of the change in weights

© eric.anquetil@irisa.fr

- Sensitivity deduce from the gradient descent  
**hidden-to-output** ( $j \rightarrow k$ ) weights

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial e_k} \frac{\partial e_k}{\partial w_{kj}} = \delta_k z_j \text{ (because } e_k = w_{kj} z_j, \text{ partial input of } a_k \text{)}$$

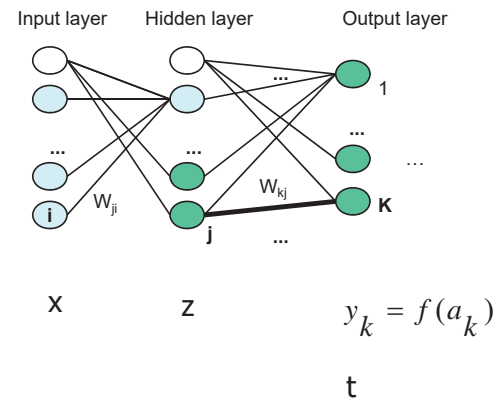
$$\delta_k = \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial e_k} = (y_k - t_k) f'(a_k)$$

$$\Delta w_{kj} = -\eta \delta_k z_j$$

$e_k = w_{kj} z_j$  : partial input of  $k$  ( $j \rightarrow k$ )

$y_k$  output of  $k$

$z_j$  output of  $j$



© eric.anquetil@irisa.fr

- Sensitivity deduce from the gradient descent at a **hidden unit** ( $i \rightarrow j$ ):
  - the sum of the individual sensitivities at the output units weighted by the hidden-to-output weights  $w_{kj}$ ; all multiplied by  $f'(a)$

$$\delta_j \equiv f'(a_j) \sum_{k=1}^c w_{kj} \delta_k$$

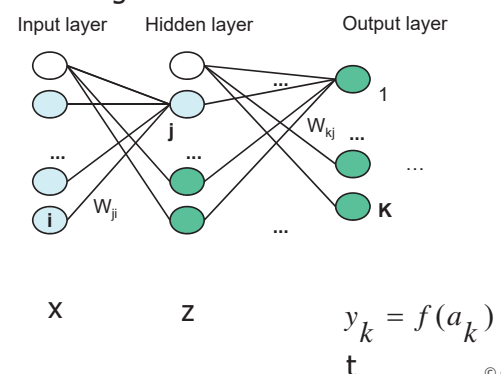
$$\Delta w_{ji} = -\eta \delta_j x_i$$

- Backpropagation algorithm

The weights are initialized with pseudo-random values and are changed in a direction that will reduce the error:

```

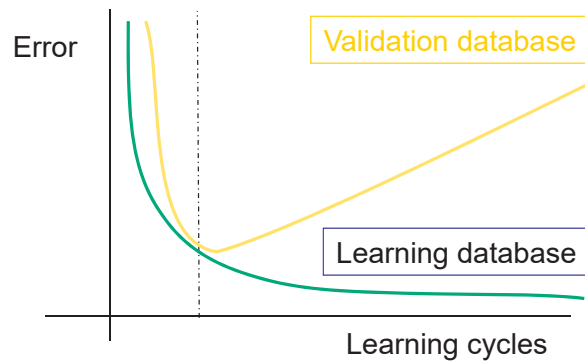
Begin  initialize       $n_H$ ;  $w$ ,  $\eta$ ,  $m=0$ 
do   $m = m + 1$ 
     $x^m \leftarrow$  randomly chosen pattern
     $w_{ji} = w_{ji} - \eta \delta_j x_i$ ;  $w_{kj} = w_{kj} - \eta \delta_k z_j$ 
until Stopping criterion
return  $w$ 
End
```



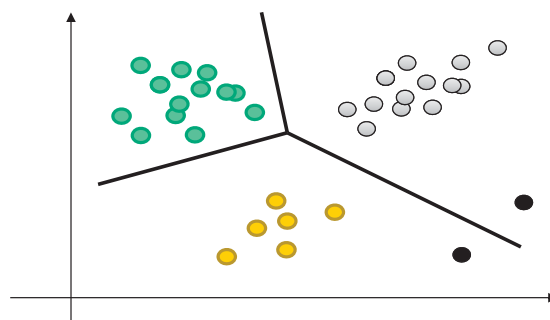
© eric.anquetil@irisa.fr

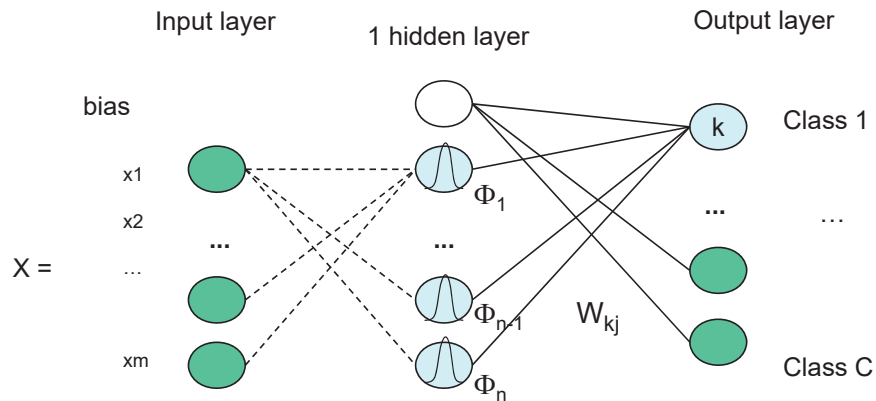


- Learning with validation (to avoid overfitting)
  - Two Learning Databases:
    - One for the learning phase
    - One for the validation of the learning
  - Test Database
    - Generalization evaluation

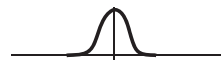


- knowledge modeling
  - Easy/Powerful learning
  - Knowledge are distributed il all the weight of network
  - Black-box system
  - Discriminative learning: with Hyper planes





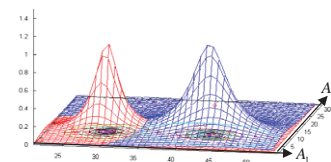
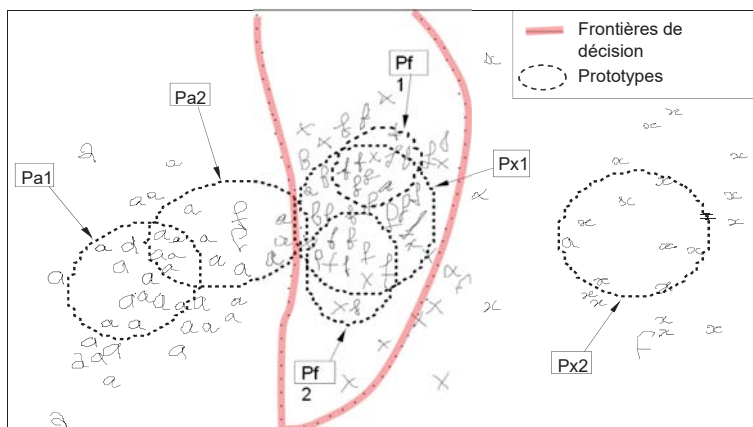
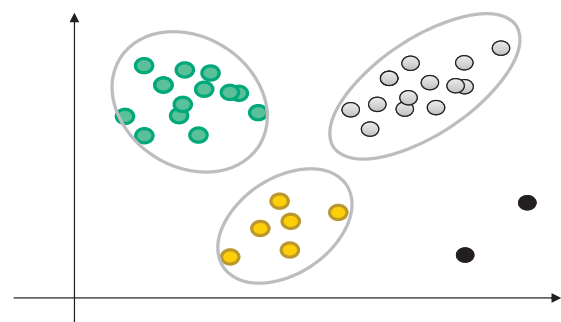
- $\Phi$  : radial activation function  
distance measure to the prototype  
(linear combination)



Output

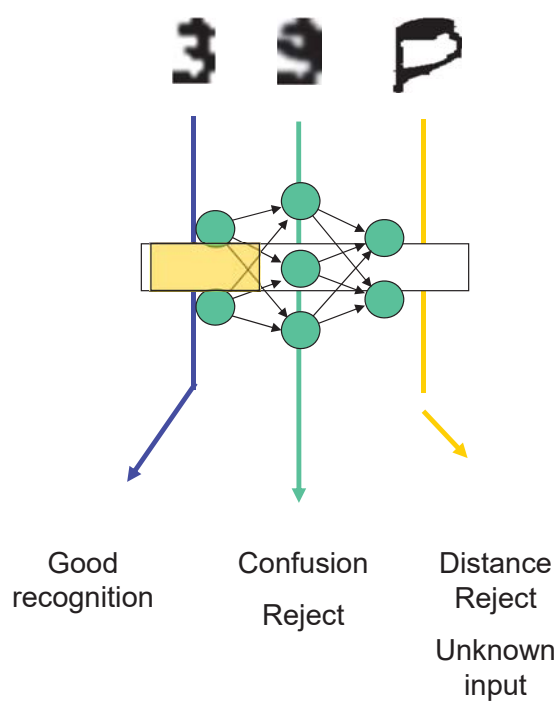
$$y_k = \sum_{i=1}^n w_{kj} \Phi_j(X) + w_{k0}$$

- Two approaches for the learning phase:
  - 1/ Globally by backpropagation
  - 2/ In two phases
    - a/ clustering to initialize the centers of the Radial Basis Function (RBF)
    - b/ Output Weights
      - learning by Least Mean Square (LMS)

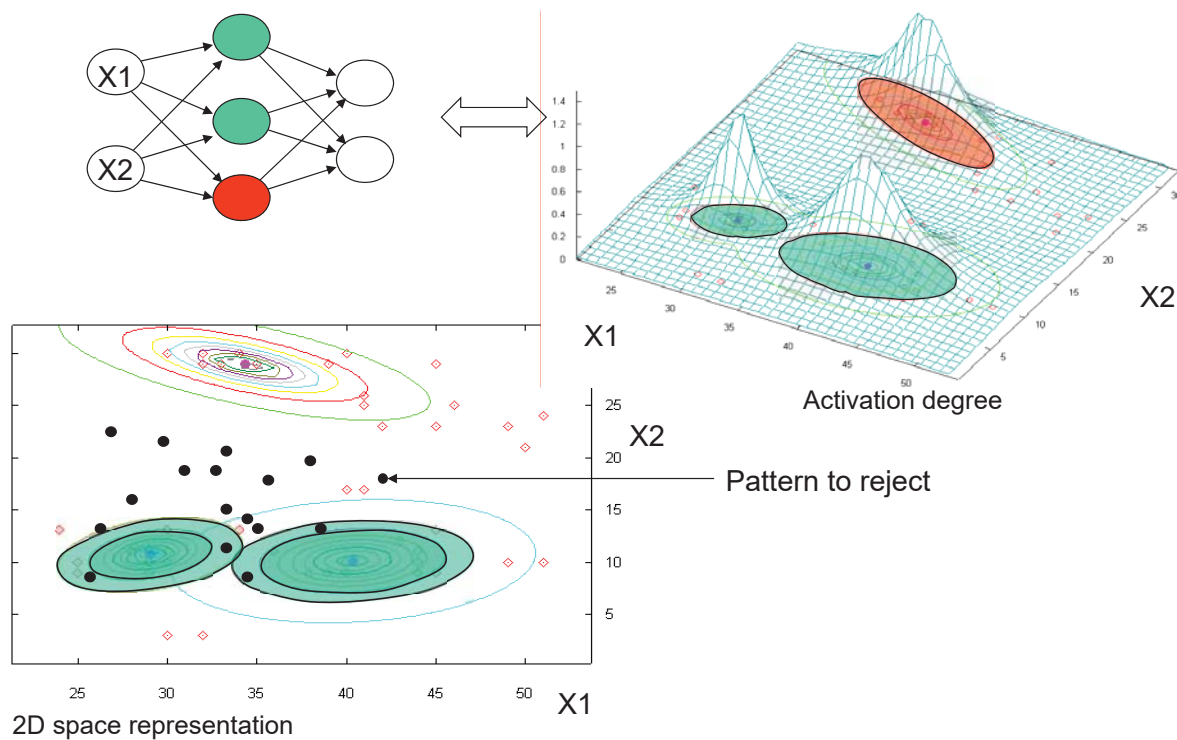
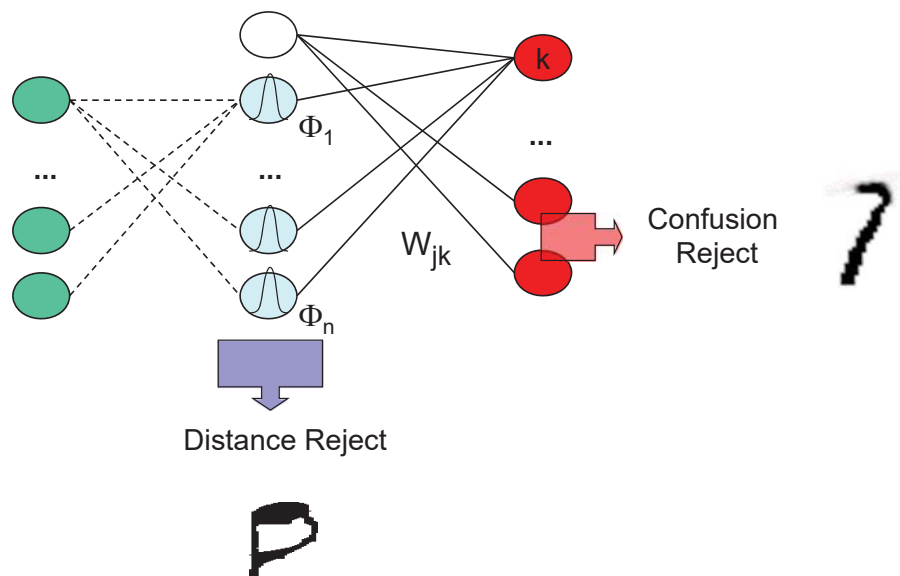


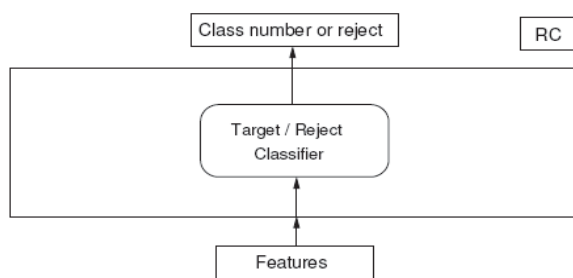
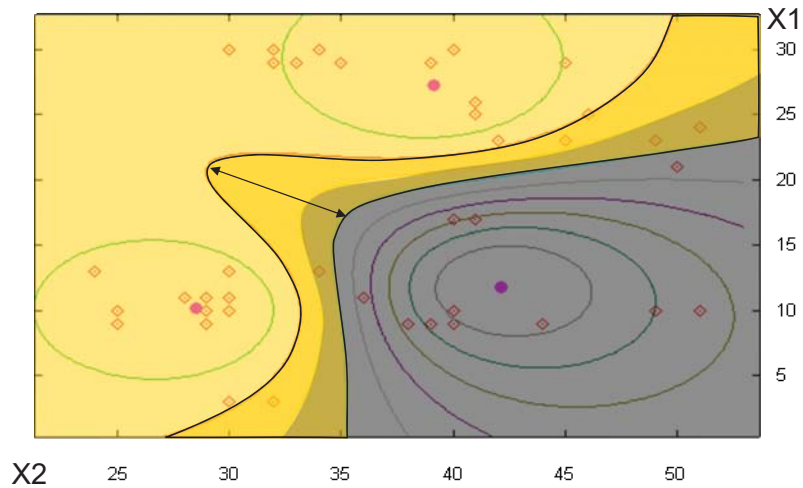
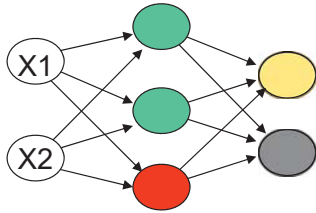
## \_Chapitre 17

### Reject Option

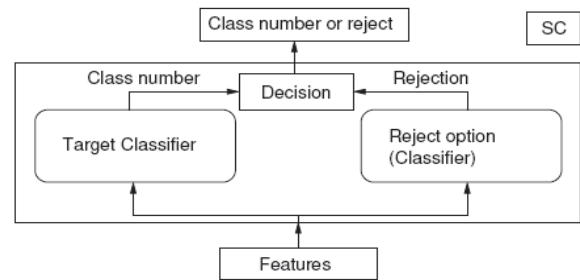


- With MLP : only confusion reject
- With RBFNN : both confusion and distance reject

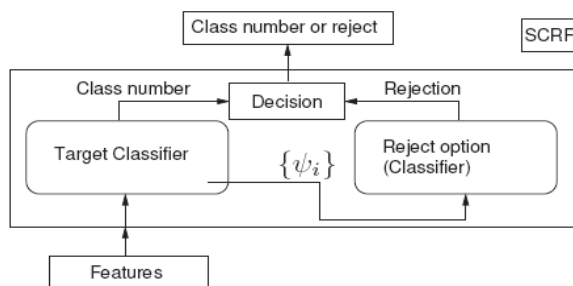




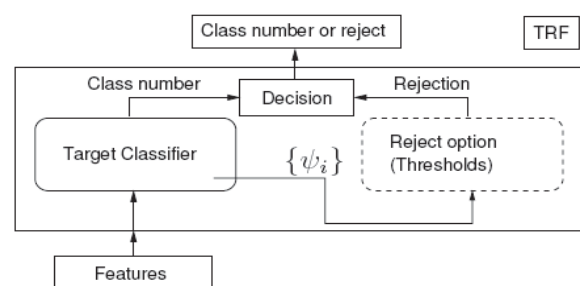
(a) a Reject Class in the target classifier



(b) a Specialized Classifier on the feature space



(c) a Specialized Classifier on the Reliability Functions  $\{\psi_i\}$



(d) Thresholds on the Reliability Functions  $\{\psi_i\}$

[Mouchère07]

■ Evaluation measure

Test Outcome		Desired Positive	Desired Negative
	Positive $N_E$	True Positive $N_E^A$	False Positive $N_R^A$
	Negative $N_R$	False negative $N_E^R$	True Negative $N_R^R$

■ Recognition/Error Rates

- TAR: True Acceptance Rate
- FAR: False Acceptance Rate

$$TAR = \frac{N_E^A}{N_E}$$

$$FAR = \frac{N_R^A}{N_R}$$

■ Accuracy Rates ("fiabilité")

- Global performance point of view

$$\text{Accuracy} = \frac{N_E^A + N_R^R}{N_E + N_R}$$

■ recall ("rappel")

- information retrieval → the number of relevant documents retrieved by a search / the total number of existing relevant documents

$$\text{Recall} = TAR$$

■ Precision ("précision")

- the number of items correctly labeled ∈ the positive class / the total number of elements labeled ∈ the positive class
- information retrieval → number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search

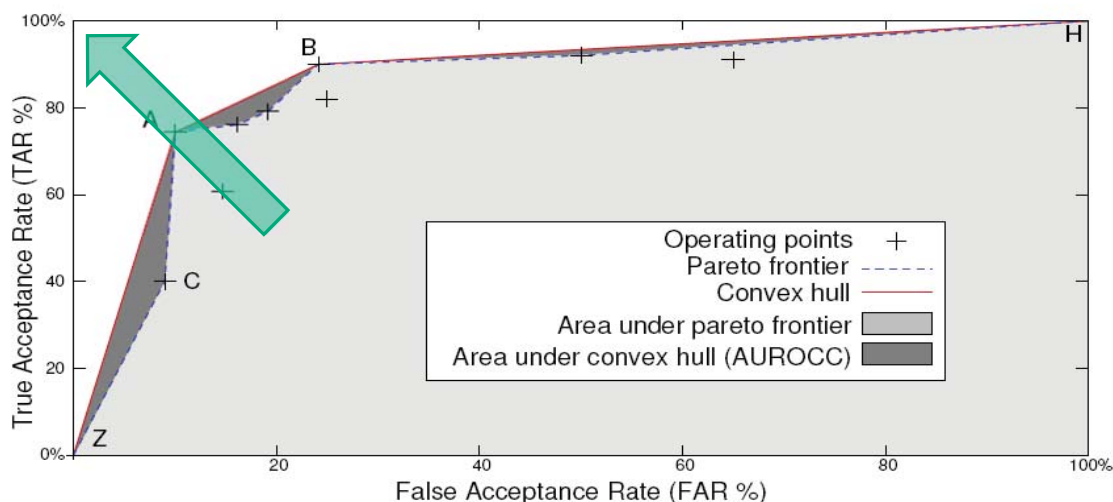
$$\text{Precision} = \frac{N_E^A}{N_E^A + N_R^A}$$

■ Evaluation of outlier(distance) rejection

- ROC curves (Receiver Operating Characteristics)
- The optimum operating point is the top left point

$$TAR = \frac{N_E^A}{N_E}$$

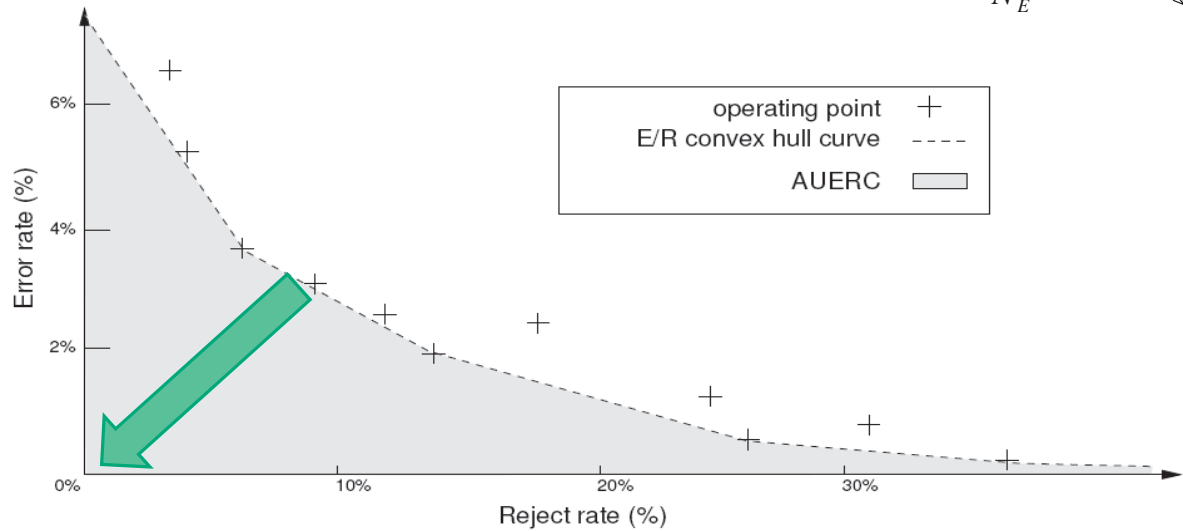
$$FAR = \frac{N_R^A}{N_R}$$



- Evaluation of confusion rejection
  - error/reject curve (E/R curve)
  - The optimum operating point is the bottom left point

$$Err = \frac{N_E - N_E^A}{N_E}$$

$$Rej = \frac{N_E^R}{N_E}$$



© eric.anquetil@irisa.fr



- Origin in statistic learning theory; class of optimal classifiers
  - Main problem of the statistic learning theory: Generalization ability
    - **When does a low training error cause a low real error?**
- Large/Max-Margin classifier / Linear Separable Classes

- With SVM a discriminating hyperplane with maximal border is searched.  
*Optimal: that with the largest of all possible discrimination planes*
- Clear reasonable (with constant intra classes variation classification confidence grows with increasing interclass distance)
- Theoretically SVM are justified by statistic learning theory

$$X = \{u_j, c_j\}_j \text{ where } c_j = \begin{cases} +1 & \text{if } u_j \in C_1 \\ -1 & \text{if } u_j \in C_2 \end{cases}$$

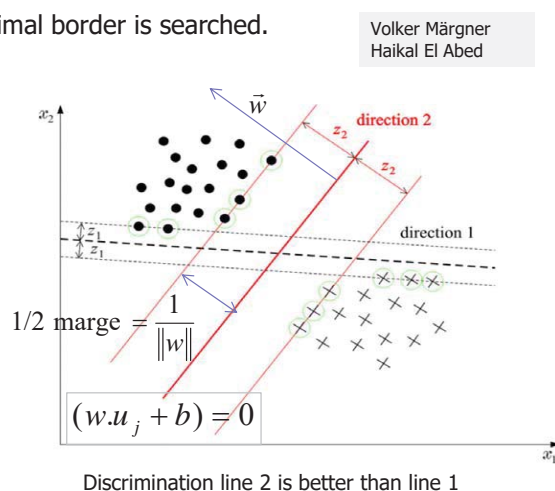
find  $w$  and  $b$  such that

$$w u_j + b \geq +1 \text{ for } c_j = +1$$

$$w u_j + b \leq -1 \text{ for } c_j = -1$$

which can be rewritten as

$$c_j (w u_j + b) \geq +1$$



© eric.anquetil@irisa.fr

- Training max-Margin classifier
  - **Constraint** optimization (two classes  $C_1$  et  $C_2$  (+1,-1))
    - To find support vector /hyperplan parameters
    - Margin to closest +1 ( $u_1$ ) and -1 ( $u_2$ ) points to be 1

$$-1(w.u_2 + b) = 1$$

$$+1(w.u_1 + b) = 1$$

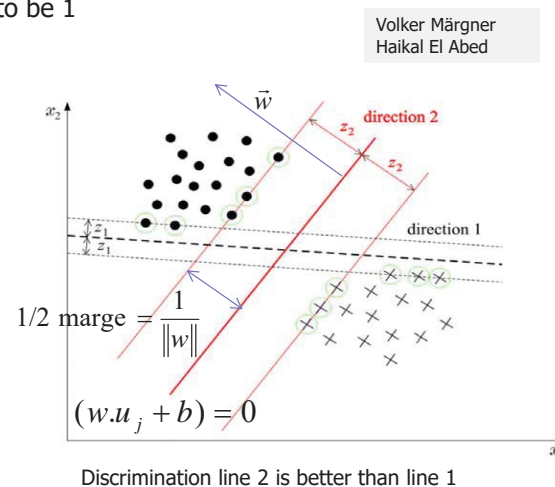
- Maximize  $\text{marge} = \frac{2}{\|w\|}$

- Minimize  $\frac{1}{2} \|w\|^2$

Maximize the margin & Vectors  $u_i$  outside the volume

$$\min \frac{1}{2} \|w\|^2 \text{ subject to } c_j (w u_j + b) \geq +1, \forall j$$

- Unconstrained problem using Lagrange multipliers



© eric.anquetil@irisa.fr

### ■ Classification

- Given unknown vector  $u$ , predict class (-1 or 1) as follows:

$$h(u) = \text{sign}\left(\sum_{i=1}^k \alpha_i y^i x^i \cdot u + b\right) = \text{sign}(w \cdot u + b)$$

- The sum is over  $k$  support vectors  $(x^i, y^i)$

### ■ If Not linearly separable (Soft Margin)

- Vectors  $u_i$  **outside** the volume, which are correctly classified ( $c_i$ ) i.e.

$$c_j(w \cdot u_j + b) \geq 1 \quad \longrightarrow \quad \xi_j = 0$$

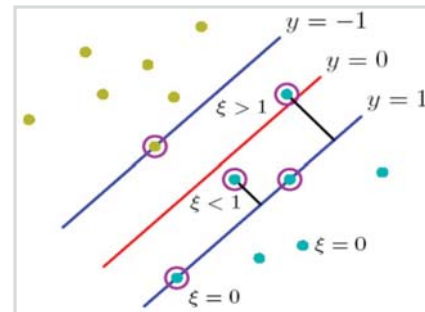
- Vectors **inside** the volume, which are correctly classified, i.e.

$$0 \leq c_j(w \cdot u_j + b) < 1 \quad \longrightarrow \quad 0 < \xi_j \leq 1$$

- Vector, which are **wrongly classified**

$$c_j(w \cdot u_j + b) < 0 \quad \longrightarrow \quad \xi_j > 1$$

- Parameter  $C$  can be viewed as a way to control overfitting: it "trades off" the relative importance of maximizing the margin and fitting the training data.



If no discrimination line exists (slack variables)

$$c_j(w \cdot u_j + b) \geq 1 - \xi_j$$

minimize

$$\frac{1}{2} \|w\|^2 + C \sum_{j=1}^m \xi_j$$

© eric.anquetil@irisa.fr

### ■ Nonlinear SVM → try a higher dimensional space

- Problem: Very high dimension of the feature space
- i.e. polynomes  $p$ -th order  $\mathbb{R}^n \Rightarrow \mathbb{R}^m, m = O(n^p)$

### ■ Advantage with SVM

- Learning depends only on dot product of sample pairs
- Recognition depends only on dot product of unknown with sample

### ■ Trick with kernel functions:

- Originally in  $\mathbb{R}^n$  :  $r$  products  $x_i x_j$
- New in  $\mathbb{R}^m$  :  $r$  product  $\Psi(x_i) \Psi(x_j)$

### ■ Solution:

- $\Psi(x_i) \Psi(x_j)$  calculated explicitly, but can be expressed with reduced complexity with kernel functions

$$K(x_i, x_j) = \Psi(x_i) \Psi(x_j)$$

### ■ Example: for the transformation

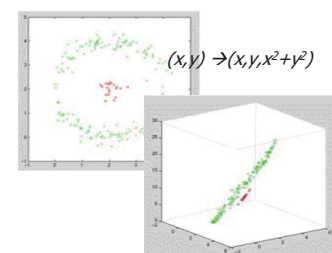
$$\Psi : \mathbb{R}^2 \Rightarrow \mathbb{R}^6$$

$$\Psi((y_1, y_2)) = (y_1^2, y_2^2, \sqrt{2}y_1, \sqrt{2}y_2, \sqrt{2}y_1y_2, 1)$$

- computes the **kernel function** the scalar product in the new feature space

$$K(x_i, x_j) = (x_i x_j + 1)^2 = \Psi(x_i) \Psi(x_j)$$

$$\mathbb{R}^6$$



© eric.anquetil@irisa.fr

■ Strengthens of SVM

- SVM supplies very **good classification** results according to present expertise; for a set of tasks it is considered as the "Top Performer"
- Sparse-representation of the solution by **support vectors**
- **Easily applicable**: small parameter set, no a-priory-knowledge necessary
- Theoretical statements about results: global optimum, generalization ability

■ Weaknesses of SVM

- **Multi-class approach** still subject of research (extension to more classes e.g. with a hierarchical procedure, where one certain class and the remainder are regarded as two classes )
- **Slow and memory-intensive learning**
- Tuning of SVMs is still a "black art": Selection of a **specific kernel** and suitable parameters is made by tests