# ESTIMATION RECURSIVE D'UNE PARTITION, EXEMPLES D'APPRENTISSAGE ET AUTO-APPRENTISSAGE DANS $R^N$ ET $I^N$

J. AGUILAR MARTÍN, M. BALSSA, R. LÓPEZ DE MANTARAS

*Nous allons présenter dans cet article un algorithme de classification du type auto-apprentissage qui peut traiter des données multidimensionnelles continues à l'intérieur du cube unitaire: CLRO.*
*La première partie traite de divers concepts généraux régissant, à notre avis, les algorithmes d'auto-apprentissage.*
*Dans une deuxième partie nous supposerons que les donnés ne peuvent prendre que les deux seules valeurs 0 ou 1; ceci nous permettra d'établir une estimation récursive d'une loi de probabilité représentée de façon exponentielle ayant une forme particulièrment simple. Nous généraliserons ensuite au cas de données réelles comprises entre 0 et 1. On montre que, moyennant certaines hypothèses simplificatrices que l'on justifie, l'algorithme conserve sa forme simple.*
*La troisième partie traite, sous le schéma général, le cas de la classification automatique de points dans l'espace $R^N$, considérés munis de mesures gaussiennes. L'algorithme est déloppé avec le souci de la simplicité des calculs en vue de son utilisation en ligne de façon récursive. Il procède à l'estimation de la moyenne et de la covariance des classes au fur et à la mesure qu'elles sont créées et modifiées selon les principes décrits pour l'auto-apprentissage. Nous donnons un exemple d'application à la reconaissance d'objects en Robotique.*

## NOTATIONS

| | |
|---|---|
| $C_k$ | classe indicée par k |
| $x$ | vecteur de composantes $x_i$, i=1,...,N |
| $P[x]$ | probabilité de l'évènement x |
| $P'[x]$ | densité de probabilité au point x |
| $P'_k[x_i]$ | densité de probabilité au point $x_i$ sachant que $x \in C_k$ |
| $p_{ik}$ | probabilité de $\{x_i=1 \mid x \in C_k\}$ |
| $q_{ik}$ | probabilité de $\{x_i=0 \mid x \in C_k\}$ |
| $\rho_{ik}$ | paramètre de densité $P'_k[x_i]$ |
| $P'[x \mid \rho]$ | densité de probabilité au point x dépendant du paramètre $\rho$ |
| $\mu_k$ | centre de la classe $C_k$ |
| $\sum k$ | matrice de dispersion (ou d'inertie) de la classe $C_k$ |
| $t_k$ | nombre d'éléments de $C_k$ |
| $\hat{\theta}_k(t_k)$ | estimation du paramètre $\theta$ correspondant à la classe $C_k$ après y avoir attribué $t_k$ éléments ($\theta$ peut être remplacé par les vecteurs p,q, , où la matrice $\sum$ ou chacune de leurs composantes). |

## 1. LES ALGORITHMES DE CLASSIFICATION PAR AUTO-APPRENTISSAGE

### 1.1 INTRODUCTION À LA NOTION D'APPRENTISSAGE

L'apprentissage est un concept très controversé et sa complexité et la place qu'il occupe dans le comportement humain dépasse largement notre propos.

De nombreux processus, comme la création d'habitudes ou "conditionnement", sont élaborés à partir du phénomène de répétition, mais, malgré leur importance nous ne les envissagerons pas car ils sortent du cadre de notre étude.

Notre intérêt se portera sur des processus d'apprentissage naturel d'un type plus "déductif" comme la mise en évidence de traits caractéristiques d'un ensemble d'objects, la déduction de relations, ou la partition d'un environement.

L'apprentissage consiste alors en la découverte, à partir des informations contenues dans l'environement, d'un concept n'apparaissant pas à priori.

Aquest treball ha estat presentat a l'INRIA (Paris) el desembre de 1980 en un Seminari sobre "Clasification Automatique et Perception par Ordinateur"

J. Aguilar Martin, M. Balssa, R. Lopez de Màntaras. Laboratoire d'Automatique et d'Analyse des Systèmes du Centre National de la Recherche Scientifique. 7, Avenue du Colonel Roche. 31400 Toulouse (France)
R. Lopez de Màntaras és actualment a la Facultat d'Informàtica de la Universitat Politècnica de Barcelona

Article rebut l'Octubre de 1981.

La présence de fonctions de perception et de mémorisation est donc nécessaire, ainsi que dans certains cas, celle d'une fonction permettant d'évaluer la qualité des décisions basées sur l'apprentissage. Nous nous trouverons donc en présence de la situation suivante:

Un ensemble d'objets $X_1 \ldots\ldots X_N$ est "examiné" séquentiellement et fournit au système une séquence d'observations $\{y_s\}_{s=t_o}^{t} = Y^t$. Le processus d'apprentissage consiste en la construction:

i) d'une estimation de l'ensemble des objets
ii) d'une relation structurelle induite par l'ensemble des objets,

les deux étant basées sur l'information passée $Y^t$; ceci permet au système:

- d'une part de reconnaître l'origine d'une nouvelle observation $y_{t+1}$, c'est-à-dire de décider de quel objet $X_i$ elle provient; cette reconnaissance peut être entâchée d'erreur,
- d'autre part de générer des objets virtuels (estimés) qui soient en acord avec la structure relationnelle déduite des observations.

La structure d'un tel système peut être représentée par le schéma de la figure 1.

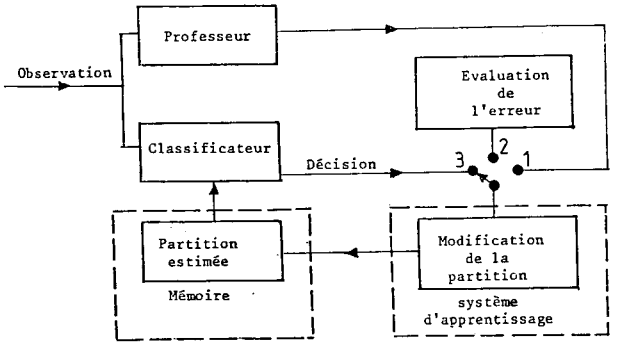



Fig. 1: Sistème de classification avec apprentissage

Un classificateur est un système, qui, étant donné un ensemble $\Omega$ affecte à una classe $C_i$ tout élément $\omega \in \Omega$. Il en résulte une partition $P = \{C_i\}_{i=1}^{N}$.

Une partition $P$ est équivalente à un ensemble de fonctions caractéristiques

$$X = \{X_i(\omega)\}_{i=1}^{N} \quad \text{où} \quad X_i(\omega) = \begin{cases} 1 \text{ si } \omega \in C_i \\ 0 \text{ si } \omega \notin C_i \end{cases}$$

à laquelle on adjoint une règle de décision $R(./P)$, ou classificateur. Une règle de décision est une application de $\Omega \to P$ avec $R(\omega/P) = C_j$ telle que: $\omega \in C_j$.

## 1.2 PRINCIPE DES ALGORITHMES PAR AUTO-APPRENTISSAGE

Les algorithmes que nous allons étudier plus particulièrment dans ce travail diffèrent de la pluplart des méthodes de classification automatique (type nuées dynamiques) /1,2/ -- par le fait que la notion d'estimation remplace celle d'optimisation. Il s'agit donc de construire, de façon séquentielle, des estimateurs d'une partition supposée inconnue mais cohérents par rapport à la métrique choisie.

L'estimation peut être elle-même basée sur une optimisation mais elle ne portera alors que sur le passé, c'est-à-dire sur l'ensemble des éléments observés. L'estimation prétend cependant, et c'est là son utilité, de perpétuer sa validité dans un futur encore inexploré. Par ailleurs, on peut remarquer qu'un estimateur est uniquement une statistique ou fonction mesurable sur la tribu des observations à laquelle on adjoint des qualités de consistence, de convergence temporelle ou d'optimalité par rapport à un critère d'erreur ou de mesure (variance ou vraisemblance).

La structure mal détérminée de l'ensemble des candidats pour ces estimateurs rend très difficile l'analyse directe du problème de classification automatique à auto-apprentissage sous forme de problème d'estimation d'une partition.

Pour vaincre cet obstacle, si une paramétrisation est possible, on ramène le problème à une estimation récursive de paramètres, auquel s'ajoute un problème de décision. Ceci le différencie, dans le cas autonome (sans professeur) d'un simple problème d'estimation statistique de paramètre.

Nous allons maintenant étudier, de façon pratique et détaillée, le fonctionement d'un tel algorithme.

Remarquons tout d'abord que le nombre de classes n'est pas imposé mais déterminé librement par le système.

Les éléments à classer sont traités sèquentiellement, une seule fois chacun, et l'algorithme peut fournir, à tout instant, une partition des éléments déjà traités sans avoir "eu connaissance" de l'ensemble des éléments à classer. Dès que le dernier élément a été traité on dispose de la partition finale. Il s'agit là de différences essentielles avec les algorithmes d'échange.

Chaque classe sera représentée par un (ou plusieurs) paramètres $\theta_k$. Afin d'alléger les notations nous parlerons, dans la suite de ce chapitre consacré aux généralités, du paramètre $\theta_k$ (au singulier) caractéristique d'une classe $C_k$ bien que certains algorithmes utilisent deux paramètres pour caractériser chaque classe.

Les différentes étapes de l'algorithme sont les suivantes:

1 - Lecture du premier élément, il est affecté à la classe 1; le paramètre $\theta_1$, représentatif de cette classe est alors calculé en fonction de cet élément;
2 - lecture de l'élément i; un calcul d'estimation de probabilité fournit alors une mesure de l'appartenance de cet élément aux différentes classes déjà créées. Cet élément est alors affecté, suivant la règle du maximum de vraisemblance, à la classe la plus probable, sous réserve que cette mesure d'appartenance à la dite classe soit supérieure à un certain seuil, sinon on crée une nouvelle classe à laquelle on affecte l'élément i;
3 - après affectation d'un élément à une classe $C_k$ le paramètre $\theta_k$, caractérisant cette classe est modifié de façon à tenir compte de la présence de ce nouvel élément dans la classe;
4 - on passe à la lecture d'un nouvel élément et on recommence les étapes 2 et 3. Après la lecture du dernier élément on dispose de la partition finale et d'une estimation des paramètres $\theta_k$.

Il est à remarquer, dans la présentation que nous venons de faire qu'aucune possibilité n'existe pour un élément de changer de classe après son affectation. La classificacion ainsi obtenue est appelée "historique".

Pourtant il se peut qu'à la fin de la classification historique un élément ayant été affecté à un instant donné à la classe $C_k$ soit plus proche d'une autre classe $C_k$, car les paramètres $\theta_k$ et $\theta_k$, ont été modifiés en cours de traitement.

Pour rémédier à cet inconvénient, on effectuera, après la classification historique une opération de classement. Pour cela on retraitera séquentiellement tous les éléments et on les affectera à la classe dont ils sont les plus proches, cette affectation étant basée sur les valeurs finales des paramètres $\theta_k$. Ainsi, la partition finale qui sera retenue sera celle obtenue après cette opération de classement que nous appellerons "classement instantané".

## 1.3 Remarques

Le principe de fonctionement de ces algorithmes appelle un certain nombre de remarques: /3/.

1. Le seuil: Il joue un rôle prépondérant. Il évite que des décisions soient prises sur des probabilités d'appartenance trop faibles. Il jouera également un rôle, comme nous le verrons plus loin, dans le nombre de classes créées. En particulier, plus sa valeur sera élevée plus le nombre de classes aura tendance à augmenter ce qui renforcera par ailleurs leur homogénéité. Le choix de ce seuil n'est pas toujours aisé. Souvent, cependant, il s'introduit de façon naturelle en fonction de la structure de l'algorithme, mais sa détermination précise conserve, malgré tout, un aspect un peu empirique. La valeur du seuil est directement liée aux caractéristiques d'une classe "vide" ou indifférenciée prête à accueillir tout élément qui n'aurait pas une appartenance suffisante à une des classes existantes.

2. Initialisation des paramètres $\theta_k$: L'initialisation des paramètres $\theta_k$ peut parfois présenter une certaine difficulté. Cela dépend de la nature de $\theta_k$. Si $\theta_k$ est une moyenne, l'initialisation se fera naturellement à partir de la valeur du premier élément affecté à la classe. Il faudra cependant que cet élément ne donne pas une "impression" trop forte

à la classe pour qu'elle soit en mesure d'accepter des éléments dont la distance soit non nulle avec ce dernier. Ceci peut être obtenu par une pondération non nulle de la valeur a priori ou d'initialisation. Si par contre, $\theta_k$ est une matrice de covariance empirique il est clair que la connaissance d'un seul élément ne permetra pas sa détermination. Il faut alors initialiser toutes les classes par une valeur $\theta_0$ que l'on doit déterminer à l'avance. Cette détermination peut parfois se révéler délicate et comporte une part d'arbitraire. Une connaissance préalable des données pourra nous guider dans ce choix.

3. Influence de l'ordre de traitement des données: puisque le paramètre $\theta_k$ est modifié à chaque arrivée d'un nouvel élément dans la classe $C_k$, et compte tenu du principe même de l'algorithme, il est clair que l'ordre dans lequel les données sont traitées pourra avoir une influence sur la partition finale. Ceci est un aspect inhérent à l'apprentissage naturel.

Ce phénomène ne présente cependant pas que des aspects négatifs. En effet il peut s'avérer intéressant de disposer de plusieurs partitions d'un même ensemble. Cet intérêt est même accru par le fait que les partitions peuvent comporter un nombre de classes différent pour una même valeur du seuil. En effet cela peut nous apporter une aide précieuse pour apprécier la validité de notre classification.

4. Nombre de classes de la partition: le nombre de classes créées n'est pas déterminé à l'avance par l'utilisateur mais fixé par l'algorithme. Cependant l'utilisateur peut imposer un nombre maximum de classes à ne pas dépasser. L'algorithme procède alors de la façon suivante:

Si la donnée à classer ne peut être affectée à aucune classe déjà créée, en satisfaisant le seuil, il cherche à créer une nouvelle classe; alors deux cas peuvent se présenter:

- si le nombre maximum autorisé de classes n'est pas encore atteint il y a création de classe
- dans le cas contraire la donnée est affectée à la classe dont elle est la plus proche mais sans modifier le paramètre $\theta_k$ relatif à cette classe; elle n'est donc pas prise en compte pour l'apprentissage et le programme signalera que le "niveau de confiance" n'a pas été atteint.

5. Fonctionement en mode professeur: si l'on a une connaissance particulière des données nous permettant de savoir, à priori, affecter certains éléments à certaines classes, on pourra utiliser cette information initiale pour fixer, dès le départ le noyau des classes. Les décisions d'affectation ne sont donc plus prises "librement" par l'algorithme mais dictées par le "professeur" dans ce cas là.

Cela donnera bien sûr une plus grande stabilité à la classification et réduira (ou éliminera) le côté un peu arbitraire que peut parfois comporter le choix de l'initialisation des paramètres $\theta_k$ en l'absence de toute information initiale.

Ce type de fonctionement est particulièrement intéressant en reconnaissance de forme.

Pour terminer ce paragraphe de généralités sur les algorithmes par auto-apprentissage nous donnons l'organigramme général d'un tel algorithme en Figure 2.

## 2. PARTITION DES SOMMETS DU CUBE UNITAIRE

Les données à classer se présentent sous la forme d'un vecteur x de N composantes dont chacune d'elles ne peut prendre que l'une des valeurs 0 ou 1. Ainsi une donnée pourra être représentée par un tableau à 2 lignes et N colonnes, une seule ligne étant en fait utilisée puisque chaque composante prend forcément un seul des deux niveaux possibles.

Dans ces conditions une classe $C_k$ sera caractérisée par une matrice $P_k$ à 2 lignes et N colonnes. Chaque élément représentera la probabilité $P_{ik}$ qu'à la ième composante $x_i$ d'une donnée $x \in C_k$ de présenter la valeur 1 ou $q_{ik}$ pour la valeur 0.

Cette matrice aura donc la forme suivante:

$$M_k = \begin{bmatrix} q_{1k} & q_{2k} & \cdots & q_{Nk} \\ P_{1k} & P_{2k} & \cdots & P_{Nk} \end{bmatrix}$$

or $\forall i \in \{1,2,\ldots,N\}$ on a:

$q_{ik} = 1 - P_{ik}$.

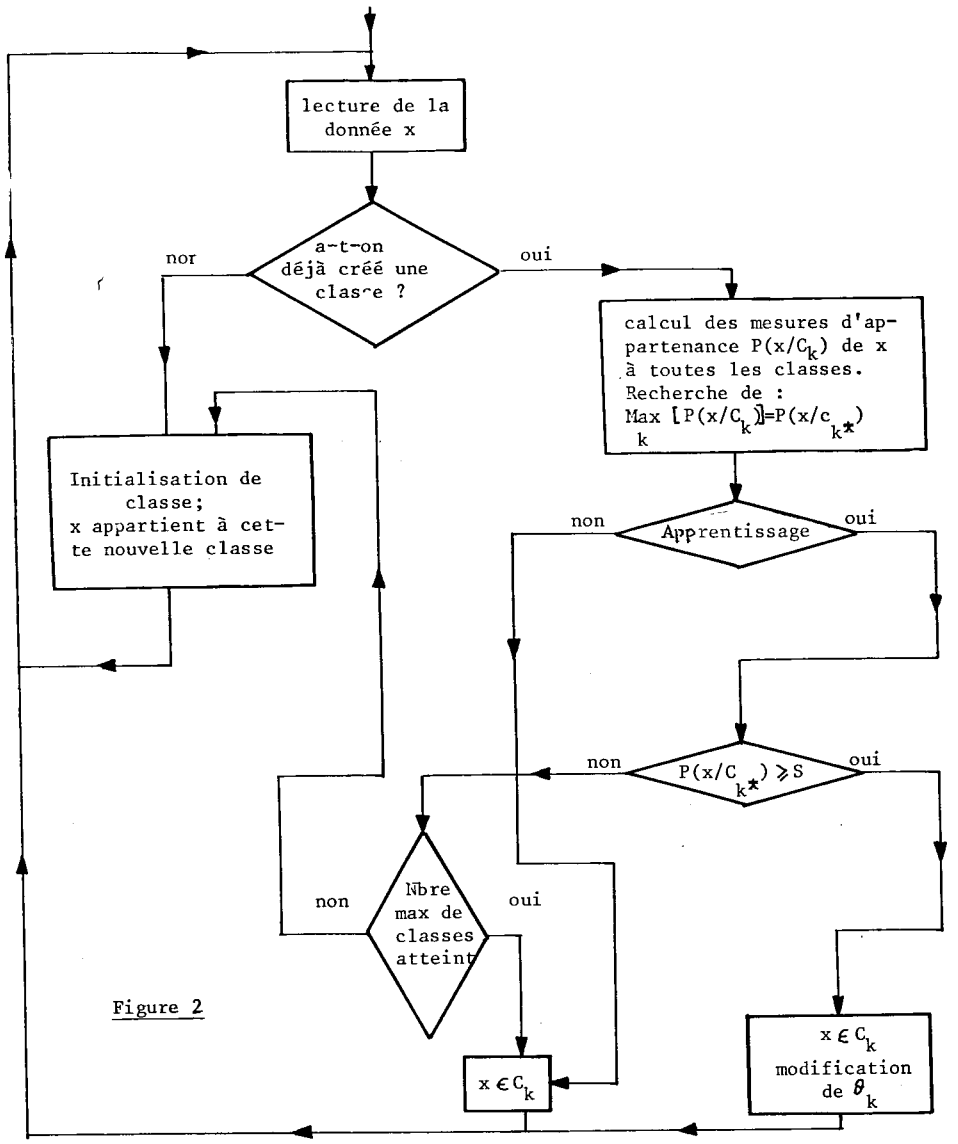



Figure 2

Figure 2: Organigramme des algorithmes de classification par auto-apprentissage.

$C_k$ pourra être entièrement déterminée par le seul vecteur ligne

$$P_k = \begin{bmatrix} P_{1k} \\ \vdots \\ P_{Nk} \end{bmatrix}$$

Soit $P_k(x)$ la probabilité que l'on observe le vecteur $x=[x_1,x_2,\ldots,x_N]^T$ sachant que cette observation provient de la classe $C_k$. En faisant l'hypothèse d'indépendance stochastique entre les composantes du vecteur x on a:

$$P_k[x] = \prod_{i=1}^{N} \text{prob}[x_i|k]$$

mais, compte tenu de la structure particulière des données $P_k(x)$ peut encore s'écrire:

$$P_k[x] = \prod_{i/x_i=1}^{N} p_{i,k} \cdot \prod_{i/x_i=0}^{N} q_{i,k} =$$

$$= \prod_{i/x_i=1}^{N} p_{ik} \cdot \prod_{i/x_i=0}^{N} (1-p_{ik})$$

Rappelons que $p_{ik}$ est la probabilité pour que la ième composante de x présente la valeur 1, sachant que x provient de la classe $C_k$.

Le paramètre $\theta_k$ défini au chapitre précédent sera donc ici représenté par le seul vecteur

$$P_k = \begin{bmatrix} p_{k,1} \\ \vdots \\ p_{k,N} \end{bmatrix}$$

La définition même des composantes de ce vecteur permet de proposer pour leur estimation d'effectuer un comptage des évènement tels que $\{x_i=1 \text{ et } x \in C_k\}$ et d'en calculer leur fréquence à l'interieur des évènements $\{x \in C_k\}$.

Définissons la par:

$$f_{ik}(t) = \frac{Nb(t, x_i=1,\ x\in C_k)}{Nb(t,\ x\in C_k)}$$

donc

$$\hat{p}_k(t) = \begin{bmatrix} f_{1k}(t) \\ \vdots \\ f_{Nk}(t) \end{bmatrix}$$

est un estimateur de $p_k$.
On peut donc poser la formule récursive suivante:

$$\hat{p}_k(t+1) = \hat{p}_k(t) + \frac{1}{t+1}\ (\delta_k(t) - \hat{p}_k(t))$$

dans laquelle

$$_k(t) = \begin{bmatrix} \delta_{1k}(t) \\ \vdots \\ \delta_{Nk}(t) \end{bmatrix}$$

et les éléments de ce vecteur sont tels que:

$\delta_{ik}(t) = 1$ si à l'instant $t, x_i=1$ et $x\in C_k$

$\delta_{ik}(t) = 0$ si $x_i=0$ et $x\in C_k$ et

$\delta_{ik}(t) = \hat{p}_k(t)$ si à l'instant $t, x\notin C_k$ pour ne pas modifier $\hat{p}_k(t)$.

Nous noterons $t_k$ le nombre de fois que, à l'instant t, on a eu $x\in C_k$ ce qui correspond au nombre d'éléments de la classe $C_k$. Remarquons que de cette façon on peut poser

$$P_{ik} = \lim_{T\to\infty} \sum_{t_k=1}^{T} x_i(t_k)$$

puisque la probabilité est la limite de la fréquence.

Nous allons tenter de conserver cette relation dans le cas où $x_i$ peut prendre toutes les valeurs dans le segment $[0,1]$.

## 3. PARTITION A L'INTERIEUR DU CUBE UNITAIRE

Nous allons, dans ce paragraphe, étendre l'algorithme au cas de données réelles continues dans le cube unitaire.

### 3.1 LOI DE PROBABILITÉ D'APPARTENANCE D'UN ÉLÉMENT À UNE CLASSE

a) Choix d'une loi de probabilité sur $[0,1]$

Dans le paragraphe précédent la loi de probabilité de la k-ième composante dépend du seul paramètre $P_k$. Ce paramètre est lui-même la probabilité de $x_i=1$. Il peut donc être interpreté comme la fréquence observée de cet évènement depuis un temps suffisamment long

Dans le cas où $x_i$ peut prendre ses valeurs dans l'intervalle $[0,1]$ on peut interpréter la valeur observée comme un degré d'appartenance de l'évènement vrai à l'évènement --- idéal $x_i=1$.

Nous poserons ainsi:

$$\rho_{ik} = \lim_{T\to\infty} \sum_{t_k=1}^{T} x_i(t_k)$$

De façon similaire au paragraphe précédent nous chercherons à utiliser, comme fonction de décision, la fonction $\rho^x(1-\rho)^{1-x}$. Cependant cette fonction ne peut pas être la loi de densité de probabilité de x sur $[0,1]$. - On doit pour cela introduire un facteur de normalisation. On aura donc:

$$p'_k(x_i) = K_i(\rho_i)\rho_i^{x_i}(1-\rho_i)^{(1-x_i)}$$

Nous verrons par la suite une justification précise du choix de cette densité.

b) Description de la loi

La densité de probabilité à la valeur - observée du vecteur x, sachant que cette observation provient de la classe $C_k$ est (en faisant l'hypothèse d'indépendance stochastique entre les N composantes du vecteur x)

$$P_k[x] = \prod_{i=1}^{N} P'_{ik}[x_i]$$

avec

$$P_{ik}[x_i] = K_{i,k}(\rho_{i,k}) \cdot \rho_{i,k}^{x_i}(1-\rho_{i,k})^{(1-x_i)}$$

où

$$\rho_{i,k} \in ]0,1[\ ;\quad x_i \in [0,1]\ ;\quad i=1,2,\ldots,N$$

$K_i(\rho_{i,k})$ est une constance de normalissation à déterminer.

Pour que $P'_{ik}[x_i]$ soit une densité de probabilité il faut que les deux conditions suivantes soient réalisées:

$$P'_{i,k}(x_i) \geqslant 0 \qquad \forall x_i \in [0,1]$$

et

$$\int_0^1 P'_{i,k}(x_i)\, dx_i = 1 \qquad i=1,2,\ldots,N$$

la 1ère condition est toujours satisfaite - puisque $x_i$ prend ses valeurs dans l'intervalle $[0,1]$.
la 2ème condition nous permettra de déterminer la constante $K_{i,k}$ qui sera donnée d'une

manière générale par:

$$\frac{1}{K(\rho)} = \int_0^1 \rho^x \cdot (1-\rho)^{(1-x)}$$

soit

$$k(\rho) = \frac{\log \frac{\rho}{1-\rho}}{2\rho-1}$$

(Dans la suite, lorsqu'il n'y aura pas d'ambiguité nous supprimerons les indices k (relatif à la classe) et i (relatif au numéro de la composante)).

Cette loi ne dépend que du paramètre $\rho$ et peut donc s'écrire:

$$P'[x/\rho] = K(\rho)\ \rho^x\ (1-\rho)^{(1-x)}$$

c) Décision bayesienne d'affectation

La mesure de l'appartenance d'un élément x à une classe $C_k$ sera la densité de probabilité à posteriori suivante: $P'[C_k/x]$. Mais une classe étant entièrement caractérisée par le vecteur $\rho$ il est équivalent d'écrire cette probabilité sous la forme: $P'[\rho/x]$ en tenant compte du fait que, bien que $\rho$ puisse prendre toutes les valeurs dans l'intervalle ouvert (0,1), seul un nombre fini de ces valeurs est à prendre en compte lors de décisions puisque le nombre de classes sera toujours défini.

Cette expression peut être calculée à partir du théorème de Bayes

$$P'[\rho/x] = \frac{P'[x/\rho].P'[\rho]}{P'[x]}$$

où:

- $P'[x/\rho]$ est la densité de probabilité conditionnelle de la mesure dont on connaît l'expression
- $P'[x]$ est la densité de probabilité marginale de x; son expression est donnée par:

$$P'\ x = \int_0^1 P[x/\rho].P'[\rho]\,d\rho = \int_0^1 P[\rho].K(\rho).\rho^x.(1-\rho)^{(1-x)}.d\rho$$

- $P'[\rho]$ est la densité de probabilité a priori d'obtenir une classe caractérisée par le vecteur $\rho$

Si on n'a aucune connaissance a priori sur le type de classe à créer il serait naturel de prendre comme loi de probabilité à priori $\rho$; la densité équiprobable

$P'[\rho] = 1$ si $\rho \in ]0,1[$

$P'[\rho] = 0$ si $\rho \leqslant 0$ et $\rho \geqslant 1$

on obtient alors:

$$P'[\rho/x] = \frac{k(\rho)\rho^x.(1-\rho)^{(1-x)}.P'[\rho]}{\int_0^1 K(\rho)\rho^x(1-\rho)^{(1-x)}.P'[\rho].d\rho}$$

expression qui, pour un x donné est proportionnelle à $K(\rho)\ \rho^x(1-\rho)^{(1-x)}$.

La présence de $K(\rho)$ dans cette formule rend difficile la comparaison entre diverses valeurs de $\rho$ pour procéder à l'atribution de x à une classe.

Nous allons voir sous quelles hypothèses cette expression peut être considérablement simplifiée.

d) Simplification

Dans le but de se ramener le plus près possible de la fonction $F(x/\rho)=\rho^x(1-\rho)^{(1-x)}$, nous allons choisir pour $P'[\rho]$ non pas la loi équiprobable mais la loi suivante qui supprimera pour $P'[\rho/x]$ l'effet de $K(\rho)$:

$$P'[\rho] = \frac{\lambda}{K(\rho)}$$

la constante de normalisation se calcule par:

$$\lambda = \frac{1}{\int_0^1 \frac{1}{K(\rho)}d\rho} = \frac{1}{\int_0^1 \frac{2-1}{\log \frac{\rho}{1-\rho}}\,d\rho}$$

Dans ces conditions l'expression:

$$P'[\rho/x] = \frac{\rho^x(1-\rho)^{(1-x)}.K(\rho).P'[\rho]}{\int_0^1 \rho^x(1-\rho)^{(1-x)}.K(\rho).P'[\rho].d\rho}$$

peut d'écrire

$$P'[\rho/x] = \frac{\rho^x.(1-\rho)^{(1-x)}.\lambda}{\lambda q(x)}$$

avec

$$q(x) = \int_0^1 \rho^x.(1-\rho)^{(1-\ )}.d\rho$$

d'où

$$P'[\rho/x] = \rho^x.(1-\rho)^{(1-x)}.\frac{1}{q(x)}$$

Ainsi, pour un x donné, cette loi est proportionnelle à la fonction $F(\rho/x)=\rho^x(1-\rho)^{(1-x)}$ Cette dernière est donc une fonction de vraisemblance pour $\rho$ lorsque x est donné.

Etude de g(x)

$$g(x) = \int_0^1 \rho^x (1-\rho)^{(1-x)} \, d\rho .$$

Cette intégrale est à rapprocher de l'intégrale classique:

$$\int_0^1 x^\alpha (1-x)^\beta \, dx$$

dont la solution est:

$$\frac{\Gamma(\alpha+1) \cdot \Gamma(\beta+1)}{(\alpha+\beta+2)}$$

En posant $\beta = 1-\alpha$ et en utilisant les deux propriétés suivantes:

$$\Gamma(1+x) = x\Gamma(x)$$

$$\Gamma(x) \cdot \Gamma(1-x) = \frac{\Pi}{\sin \Pi x}$$

on obtient

$$g(x) = \int_0^1 \rho^x (1-\rho)^{(1-x)} dx = \frac{x(1-x)\Pi}{2 \sin \Pi x}$$

La fonction g(x) ne joue pas de rôle lors du choix de ρ maximisant la probabilité à posteriori P[(ρ/x)] . Cette recherche de ρ rendant P[ρ/x] maximum s'effectuera donc sur la fonction:

$$F[\rho/x] = \rho^x (1-\rho)^{(1-x)}$$

Cette remarque réduit considérablement la complexité de l'algorithme et le gain en temps calcul qui en découle est appréciable. L'intérêt d'avoir choisi comme loi de probabilité à priori pour ρ la loi P' $\rho = \frac{\lambda}{K(\rho)}$ apparaît clairement maintenant.

e) Justification du choix de la loi

$$P'[\rho] = \frac{\lambda}{K(\rho)}$$

Le choix de cette loi (à la place de la loi équiprobable) peut être justifié au vu de sa représentation graphique. Rappelons que les conditions de normalisation s'écrivaient:

$$\lambda = \frac{1}{\int_0^1 \frac{2\rho-1}{\log \rho/1-\rho} \, d\rho}$$

et pour la loi marginale de x:

$$\lambda = \frac{1}{\int_0^1 \frac{\Pi x(1-x)}{2 \sin \Pi x} \, dx}$$

l'intégration numérique de ces expressions nous fournit la valeur de la constante λ,

λ = 2,347

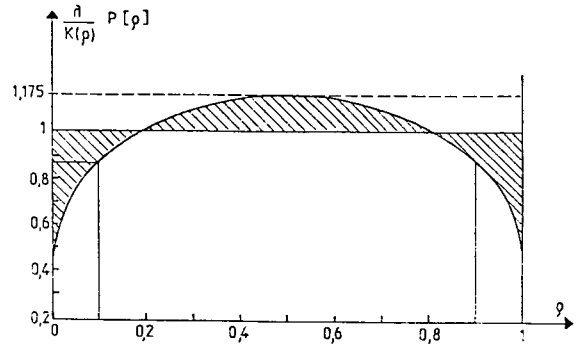



Fig. 3: Loi de probabilité a priori pour

La figure 3 donne la représentation graphique de la fonction:

$$\frac{\lambda}{K(\rho)} = \frac{2,347(2\rho-1)}{\log \frac{\rho}{1-\rho}}$$

et permet de constater que par rapport à la loi équiprobable on favorise légèrement la zone voisine de ρ=0,5 et l'on pénalise les zones où ρ est voisin de 0 ou de 1 (pour ρ=0,1 la fonction vaut déjà 0,87). Ainsi, le programme qui mettra en oeuvre cet algorithme utilisera comme mesure d'appartenance d'une donée à une classe la fonction

$$F[\rho/x] = \rho^x (1-\rho)^{(1-x)}$$

Cette fonction sera étudiée au paragraphe suivant dans le cas scalaire.

L'affectation d'une donnée à une classe s'effectuera suivant la règle du maximum de vraisemblance pour les seules valeurs $_k$ correspondant aux classes existantes.

$$x \in C_{k^*} \quad \text{si} \quad F[\rho_{k^*}/x] = \max_k (F[\rho_k/x]) \geq S$$

L'introduction du seuil S a pour but d'éviter une décision d'affectation s'appuyant sur des probabilités d'appartenance trop faible. Nous reviendrons en détail ultérieurement sur cette notion de seuil.

f) Etude de la fonction $F(\rho/x)=\rho^x(1-\rho)^{(1-x)}$ en fonction de ρ (fonction de vraisemblance

Dans la figure 4 on peut remarquer les carac

téristique de la fonction $F(\rho/x)=\rho^x(1-\rho)^{(1-x)}$ que nous justifions das ce paragraphe. Calculons la dérivée par rapport à $\rho$

$$\frac{\partial F(\rho/x)}{\partial \rho} = x\rho^{(x-1)}(1-\rho)^{(1-x)} - \rho^x(1-x)(1-\rho)^{-x}$$

d'où l'on tire

$$\frac{\partial F(\rho/x)}{\partial \rho} = \frac{\rho}{1-\rho}^{x}\left(\frac{x}{\rho}-1\right)$$

ler cas: x est à l'intérieur de l'intervalle [0,1]

$$\frac{\partial F(\rho/x)}{\partial P} = 0 \quad \text{pour} \quad \rho=x$$

la fonction aura son maximum pour $\rho=x$; il vaut A(x)

$$A(x) = x^x(1-x)^{1-x}$$

- si $\rho\to 0$

$$\frac{\partial F(\rho/x)}{\partial \rho}\ \rho^x \cdot \frac{x}{\rho} = \frac{x}{\rho^{(1-x)}}$$

puisque $x<1$

donc sa tangente est l'axe vertical à la valeur

$F(0/x) = 0$

- si $\rho\to 1$ il est immèdiat que la fonction tend vers moins l'infini donc la verticale au point $F(1/x)=0$ est sa tangente.

2ème cas: valeurs particulières aux limites pour $\bar{x}$

$x = 0 \Longrightarrow F(\rho/0) = 1-\rho$

$x = 1 \Longrightarrow F(\rho/1) = \rho$

La dérivée A'(x) du maximum est:

$$A'(x)=A(x)\text{Log}\frac{x}{1-x} = (1-x)\text{Log}\frac{x}{1-x} \cdot e^{x.\text{Log}\frac{x}{1-x}}$$

Cette quantitè s'annule pour x=1/2, donc A(x) sera minimum pour x=1/2.

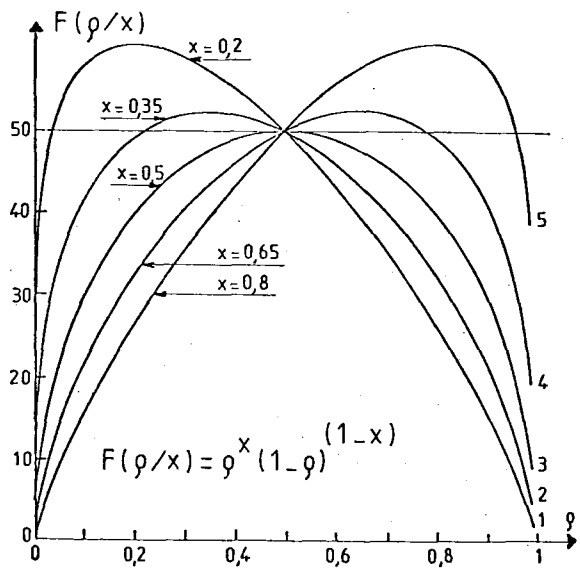



Fig. 4: Fonctions de vraisemblance

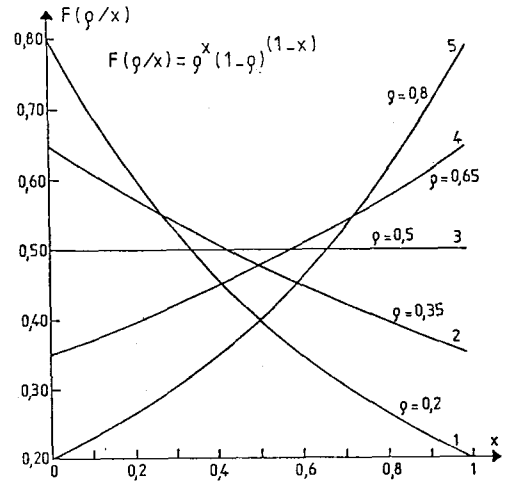



Fig. 5: Fonctions de densité

a) Etude de la fonction $F(\rho/x)$ en fonction de x (fonction de densité)

On constate aisément que la fonction est monotone en x et que pour $0<\rho<0,5$ la fonction est décroissante alors que pour $0,5<\rho<1$ elle est croissante. La valeur $\rho=0,5$ correspond à une constante quelque soit x

$F(0,5/x) = 0,5$

Les valeurs extrêmes $\rho=0$ et $\rho=1$ sont à exclure.

La figure 5 montre différentes formes de cette fonction selon les valeurs de $\rho$. Dans la figure 6 nous montrons la forme que prend la courbe pour les très petites valeurs de $\rho$.

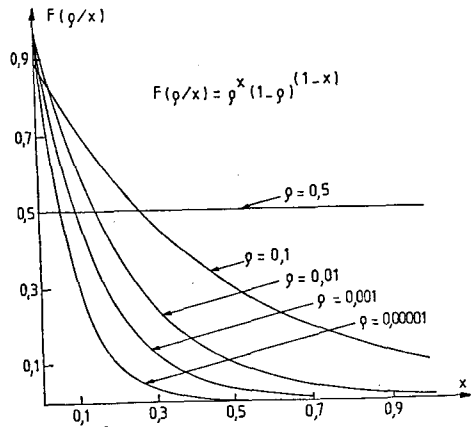



Fig. 6: Fonctions de densité pour valeurs très petites de $\rho$

## 3.2 MISE EN OEUVRE DE L'ALGORITHME

a) Détermination du seuil

En l'absence de toute information a priori on initialisera par une classe vide caractérisée par la probabilité a priori de $x_i$, - $P'_o[x_i]$ définie au moyen du paramètre $\rho_{i,o}$. On fera l'hypothèse que toutes les valeurs - de l'intervalle [0,1] sont équiprobables ce qui implique $\rho_{i,o}=1/2$. Il apparaît ainsi, de façon naturelle un seuil:

$$S(x) = \prod_{i=1}^{N} \rho_{i,o}^{x_i} \cdot (1-\rho_{i,o})^{(1-x_i)} \cdot \frac{1}{q(x_i)}$$

$$i = 1,2,\ldots,N$$

soit:

$$S(x) = \left(\frac{1}{2}\right)^{\sum x_i} \cdot \left(\frac{1}{2}\right)^{(N-\sum x_i)} \cdot \prod_{i=1}^{N} \frac{1}{q(x_i)} =$$

$$= \left(\frac{1}{2}\right)^{N} \cdot \prod_{i=1}^{N} \frac{1}{q(x_i)}$$

Ainsi l'affectation d'une donnée à une classe non vide ne pourra s'effectuer qu'avec, - pour la fonction $P'[\rho/x]$ une valeur supé--- rieure à

$$S(x) = \frac{1}{2}^{N} \cdot \prod_{i=1}^{N} \frac{1}{q(x_i)} \quad ;$$

comparar $P[\rho/x]$ à $S(x)$ revient à comparer -- $F[\rho/x]$ à $S'(x)=(1/2)^N$; en effet, le terme -- $1/q(x)$ présent dans les deux expressions --- n'intervient pas lors de la comparaison.

b) Estimation du paramètre $\rho$

Dans ce paragraphe, afin d'alléger les - notations, nous ne porterons pas l'indice de la classe et le numéro de la composante des vecteurs x et $\rho$. L'estimation de $\rho$ est ba-- sée sur un nombre q d'observations disponi-- bles, l'indice supérieur correspond à leur - ordre d'arrivée $\{x^1,x^2,\ldots,x^q\}$, si nous choi sissons l'estimateur du maximum de vraisem-- blance on a:

$$P'[\hat{\rho}/x^1,x^2,\ldots,x^q] = \max_{\rho} P'[\rho/x^1,x^2,\ldots,x^q]$$

or en faisant l'hypothèse d'indépendance des différentes observations on a:

$$P'[\rho/x^1,x^2,\ldots,x^q] = \prod_{i=1}^{q} P'[\rho/x^i]_{i=1,2,\ldots,q}$$

soit

$$P'[\rho/x^1,x^2,\ldots,x^q] = \prod_{i=1}^{q} \rho^{x^i}(1-\rho)^{(1-x^i)} \frac{1}{q(x^i)} =$$

$$= \rho^{\left(\sum_{i=1}^{c} x^i\right)} \cdot (1-\rho)^{\left(q-\sum_{i=1}^{c} x^i\right)} \cdot \prod_{i=1}^{q} \frac{1}{q(x^i)}$$

Posons: $Z = \dfrac{\sum_{i=1}^{c} x^i}{q}$ et $G = \prod_{i=1}^{q} \dfrac{1}{q(x^i)}$

$$P[\rho/x^1,x^2,\ldots,x^q] = \rho^{qz} \cdot (1-\rho)^{q(1-z)} \cdot G =$$

$$= [\rho^z \cdot (1-\rho)^{(1-z)}]^q \cdot G$$

l'estimateur $\hat{\rho}^q$ sera donc la valeur de $\rho$ telle que

$$\frac{\partial}{\partial \rho} P'[\rho/x^1,x^2,\ldots,x^q] = G \cdot \frac{\partial}{\partial \rho}[\rho^z (1-\rho)^{(1-z)}]^q = 0$$

soit:

$$\frac{[\rho^z(1-\rho)^{(1-z)}]^{q-1}}{q-1} \cdot \left(\frac{\rho}{1-\rho}\right)^z \cdot \left(\frac{z}{\rho} - 1\right) = 0$$

$\rho$ prenant ses valeurs dans l'intervalle [0,1] la seule valeur de $\rho$ satisfaisant cette relation est: $\rho=z$, par suite:

$$\hat{\rho}^q = \frac{1}{q} \sum_{i=1}^{q} x^i$$

L'estimateur de $\rho$ est donc la moyenne arithmétique des observations ce qui confirme --- l'extension de la loi obtenue pour les ----- $x_i \in \{0,1\}$.

c) Procédure d'apprentissage

La procédure d'apprentissage sera la suivante:

Pour chaque donnée x à classer on cherchera $k^*$ tel que:

$$P'_{k^*}[\rho_{k^*}/x] = \max_{k} P'_k[\rho_k/x]$$

deux cas peuvent alors se présenter:

i) si $P'_{k^*}[\rho_{k^*}/x] \geq S(x)$

alors on atribue x à la classe $C_{k^*}$ et son paramètre $\rho_{k^*}$ est modifié de la façon suivante: soit $\rho_k^t$ la moyenne de la classe $C_k$ lorsqu'elle a absorbé t éléments.

Il faut noter que $t_k$ ne correspond pas à un - temps commun pour les différentes classes. Il s'agit du nombre d'éléments de la classe k. - On ne mettra pas l'indice k pour alléger -- les notations dans ce qui suit.

On peut écrire pour la classe venant d'attirer un élément

$$\hat{\rho}(t) = \frac{x^1+x^2+\ldots+x^t}{t}$$

si $x^i$ est la ième donnée absorbée par la classe $C^k$

$$\hat{\rho}(t+1) = \frac{x^1+x^2+\ldots+x^t+x^{t+1}}{t+1} = \frac{t\hat{\rho}(t)\ x^{(t+1)}}{t+1}$$

en ajoutant et retrachant la quantité $\frac{1}{t+1}\ \rho^t$ on obtient:

$$\hat{\rho}(t+1) = \hat{\rho}(t) + \frac{1}{t+1}\ (x^{t+1} - \hat{\rho}(t))$$

formule qui constitue la forme récurrente de correction de l'estimateur du paramètre.

ii) si $P'_k[\rho_k/x] < S(x)$

il faut alors examiner si le nombre maximum autorisé pour la création de classes est atteint:

- s'il ne l'est pas on crée une nouvelle classe initialisée à partir de la donnée x
- s'il l'est on attribue x à la classe $C_{k^*}$ mais on ne modifie par le paramètre $\rho_{k^*}$ et le programme signale: "seuil de confiance non atteint".

On voit donc le rôle prépondérant joué par le seuil S(x) dans le mécanisme de croissance du nombre de classes. En particulier on remarquera que plus S(x) sera choisi grand, plus le nombre de classes créées aura tendance à augmenter.

Lors de l'établissement de la formule récurrente d'apprentissage la valeur de $x^1$ est en realité fonction du mode d'initialisation choisi; si on initialise par l'intermédiaire de la classe neutre ($\rho_o$=1/2), $x^1$ est alors la moyenne entre la valeur de la 1ère donnée et la valeur 1/2. Ce problème sera abordé en détail plus loin.

Examinons la formule de correction du paramètre

$$\hat{\rho}(t+1) = \hat{\rho}(t) + \frac{1}{t+1}\ (x^{t+1} - \hat{\rho}(t))$$

En posant $\alpha$=1/t+1 cette formule s'écrit:

$$\hat{\rho}(t+1) = \hat{\rho}(t).(1-\alpha)+\alpha x^{t+1}$$

si, au lieu de donner à $\alpha$ la valeur 1/t+1 on lui impose une valeur constante $0<\alpha<1$ on modifiera la loi d'apprentissage, en particulier:

- si $\alpha$ est petit (voisin de 0) la moyenne évolue très peu lorsque la classe absorbe de noveaux éléments; à la limite ($\alpha$=0) la moyenne reste égale à la 1ère valeur ---- qu'elle prend.
- si $\alpha$ est gran (voisin de 1) le "poids" du dernier élément absorbé par la classe est prépondérant dans le calcul de la moyenne; à la limite ($\alpha$=1) la moyenne prend la valeur du dernier élément
- si $\alpha$=1/2 le dernier élément a le même --- "poids" que tout le passé. Dans ce cas où $\alpha$=1/t+1 on dit que la loi d'apprentissage est statistique, le poids du dernier élément diminue au fur et à mesure que la classe absorbe des éléments.

d) Mécanisme de création de classes

Nous allons montrer qu'il existe une possibilité de création de classes sur tout --- l'intervalle [0,1] c'est-à-dire que $\rho$ peut prendre n'importe quelle valeur de cet intervalle.

Supposons que la première donnée à classer ait la valeur 0. La première classe sera --- alors initialisée par la valeur $\rho'$=(0.5+0)/2 =0.25; la figure 7 représente la courbe ---- $\rho^x(1-\rho)^{1-x}$ pour $\rho$=0.25. Cette courbe coupe la droite S(x)=0.5 au point $\alpha_1$. Pour qu'il y ait création d'une nouvelle classe il faut que la nouvelle donnée soit supérieure à $\alpha_1$ (sinon elle sera affectée à la classe 1); -- alors une nouvelle classe sera initialisée par $\rho^1_2$=(0.5+$x^2$)/2 déterminant un point $\alpha_2$). Il suffit donc que l'on ait $x>\alpha_i$ pour que la création de classe se poursuive jusqu'à obtenir $\rho^1_n$=0.5. Si, par contre, la 2ème donnée est inférieure à $\alpha_1$, $\rho_1$ sera modifié en l'occurence diminué. On peut se demander s'il n'existe pas de limite à la création de classes si les données à traiter sont de plus en plus petites, autrement dit si $\alpha_i \to 0$ lorsque $\rho \to 0$ (puisque la création de classe ne peut intervenir que pour $x>\alpha_i$). L'intersection des courbes $\rho^x(1-\rho)^{1-x}$ et de la droite S(x)=1/2 s'écrit:

$$\rho^\alpha(1-\rho)^{1-\alpha} = 1/2$$

Soit en prenant le Log des 2 membres:

$$\alpha = \frac{\text{Log}\ \frac{1}{2(1-\rho)}}{\text{Log}\ \frac{\rho}{1-\rho}}$$

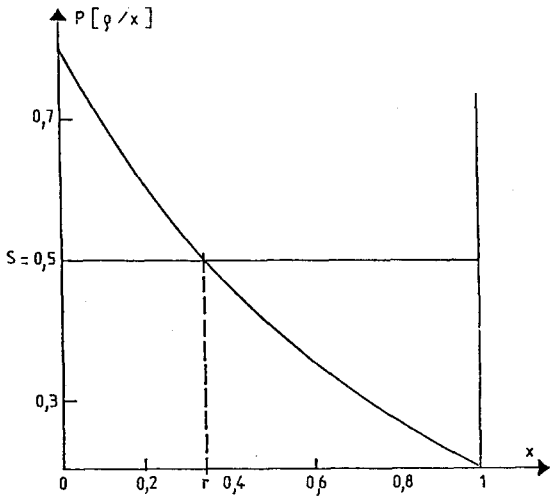



Fig. 7: Fonctions de densité pour =0.25

si $\rho \to 0 \quad \alpha \to \frac{\text{Log } 1/2}{\text{Log } 0} \to \frac{-a}{-\infty} \to 0$ par valeurs positives

il n'y a donc pas de limite inférieure théorique, à une donnée pour créer une classe. - Par symétrie on voit qu'il en est de même -- sur le segment [0'5,1]. Cependant, en pratique il est clair que la création de classes sera bien plus facile pour des valeurs voisines de 1/2 que pour des valeurs de x voisines de 0 ou de 1. On a eu la confirmation expérimentale de ce fait. On va proposer un -- certain nombre de solutions pour uniformiser la sensibilité à l'intérieur du domaine.

## 3.3 UNIFORMISATION DE LA SENSIBILITÉ À L'INTÉRIEUR DU CUBE UNITAIRE

a) Linéarisation des axes de l'espace des - données

L'équation du paragraphe précédent donne la répartition des points $\alpha_i$ intersection - de la droite S(x)=0.5 avec les courbes ----- $\rho^x(1-\rho)^{(1-x)}$; la figure 8 représente cette - courbe.

Cette non linéarité est une des causes de la moins sensibilité de l'algorithme pour les - valeurs de x voisines de 0 ou 1.

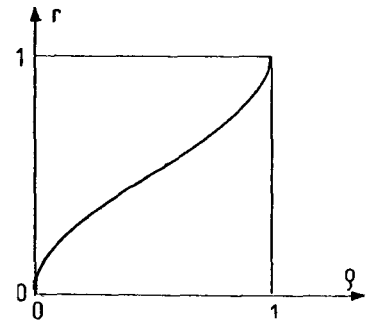



Fig. 8: Repartition des points d'intersection de $^x(1-\ )^{1-x}$ avec S=0.5

Nous avons donc procédé à une linéarisation des données. Il s'agit de multiplier chaque donnée par la fonction inverse de $\alpha(\rho)$. Malheureusement cette fonction inverse ne s'exprime pas analytiquement ce qui nous conduit à adopter une méthode numérique consistant à réaliser un tableau faisant correspondre à - chaque valeur de $\alpha$ la valeur correspondante de $\rho$.

c) Mode d'initialisation des classes

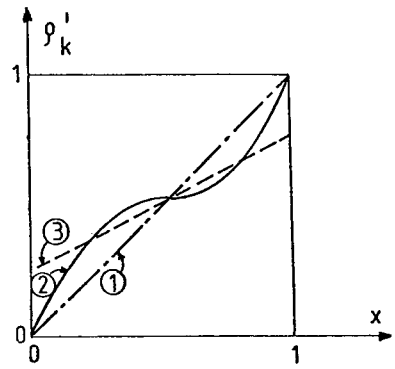
La solution retenue était d'initialiser une classe par $\hat{\rho}_k(t)=(0.5+x^1)/2$. Supposons - que la donnée $x^1$ ait la valeur 0; $\hat{\rho}_k(t)=0,25$ Ceci, comme on vient de la voir, rend très - difficile la création de classes dont le paramètre $\rho$ serait inférieur à 0.25.

Une solution consisterait à initialiser la - classe par la valeur de la première donnée - qui lui est affectée. Malheureusement, si -- cette solution est satisfaisante pour les -- valeurs de x voisines de 0ou de 1 elle rend prohibitif le nombre de classes créées vers le centre du domaine ($0.4<\rho<0.6$).

Nous avons adopté une solution intermédiaire qui tente de concilier au mieux les deux aspects contradictoires ci-dessus.

- pour $0<x\leq 0.5$ on initialise par $\hat{\rho}_k(1) = 1/2 \sin \Pi x$
- pour $0.5<x<1$ on initialise par $\hat{\rho}_k(1) = 1/2 \cos \Pi(x+1/2)+2$

La figure 9 montre la solution adoptée




① initialisation par $\hat{\rho}_k(1)=x$
② initialisation par $\hat{\rho}_k(1)=0.5+x/2$
③ solution adoptée

Fig. 9: Modes d'initialisation

c) Action sur le seuil de création de classes

Pour favoriser la création de classes -- dans la région voisine de 0 et 1 et pour la limiter vers le centre de l'intervalle [ 0,1] on a introduit un seuil variable en fonction de x et non plus constant et égal à 1/2. En effect rappelons que la loi choisie pour --- P' [ $\rho$ ] en remplacement de la loi equiprobable avait une forme (figure 3) qui favorise donc la création de classes au voisinage de $\rho=0,5$ et la limite pour $\rho$ voisin de 0 et 1.

Pour compenser le choix de cette loi nous -- avons adopté pour le seuil variable une loi particulièrement simple puisqu'il s'agit --- d'un cercle centré sur la droite $\rho=1/2$. L'or donnée du centre et le rayon du cercle peu-- vent être déterminés de façon empirique par l'utilisateur selon son désir de favoriser - plus ou moins la création de classes dans -- les régions voisines de 0 et 1. La figure 10 illustre le rôle de ce seuil variable.

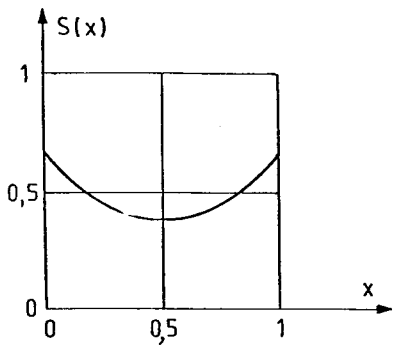



Fig. 10: Variabilité du seuil

d) Exemple

L'exemple suivant permet de constater - les progrès obtenus quant à l'uniformisation de la sensibilité à l'intérieur du carré unitaire (cas bidimensionnel), grâce aux ac---- tions décrites dans le paragraphe précédent. Il s'agit de classer 127 points de $R^2$ répartis de la façon suivante: 5 groupes "com---- pacts" et de deux points A et B situés sur un des sommets du carré.

Un premier passage de l'algorithme CLRO sans les améliorations décrites a donné les résul tats de la figure 11:

- les groupes 1,2 et 3 ont été mis dans -- une même classe
- le groupe 4 constitue une 2ème classe -- avec le point B
- le groupe 5 a été 'éclaté" en 4 classes dont l'une contient le point A.

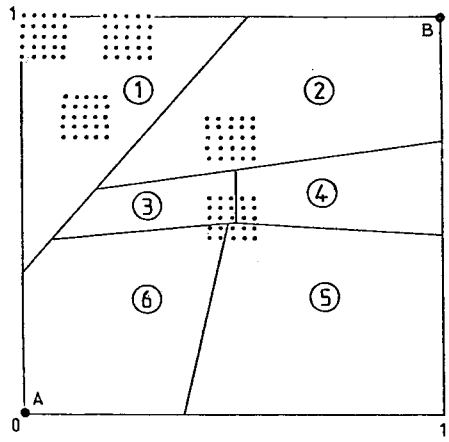



Fig. 11: Resultat de la classification sans améliorer l'algorithme

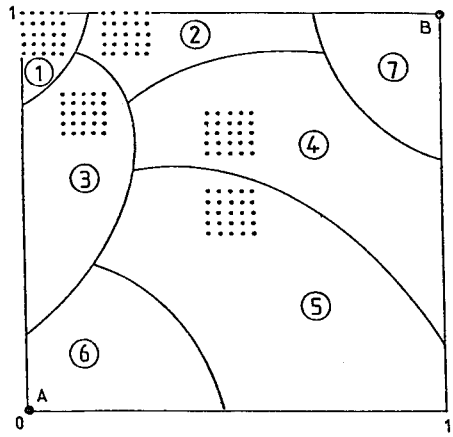



Fig. 12: Resultat de la classification avec l'algorithme amélioré

Ceci confirme donc la très forte sensibilité dans le centre du domaine et la faible sensibilité dans les régions voisines des sommets.

Un deuxième passage avec l'algorithme amélioré a donné les résultats de la figure 12:
- chacun des 5 groupes a constitué une classe
- le point A a formé une classe
- le point B a également formé une classe

Cet exemple permet donc d'apprécier la meilleure uniformisation de la sensibilité obtenue par les modifications décrites au paragraphe précédent.

## 4. PARTITIONS DANS $R^N$ MUNI DE MESURES GAUSSIENNES

### 4.1 HYPOTHÈSES SUR L'ENSEMBLE À CLASSER

Nous présentons dans ce chapitre un algorithme de classification itérative par auto-apprentissage considérant des données issues de populations gaussiennes. Son schéma général de fonctionnement est analogue au précédent (CLRO) seules diffèrent les hypothèses sur la distribution probabiliste attribuée aux données à classer. Cet algorithme, qui tient compte explicitement des corrélations entre variables, etablit l'estimation récursive des deux premiers moments de densités de probabilité gaussiennes.

Un algorithme de ce type, développé par R. López de Màntaras /7/ utilisait le filtrage linéaire adaptatif (filtre de Kalman-Bucy) pour estimer ces deux premiers moments.

L'algorithme présenté ici est considérablement plus simple et a des perfomances comparables.

Nos allons caractériser (ou paramétriser) une classe par sa moyenne (ou centre de gravité) et son ellipsoïde de dispersion (ou d'inertie) donné par la matrice de covariance . L'ensemble de ces deux paramètres est appelé le "noyau" de la classe. Puisqu'on travaille sous des hypothèses de normalité les deux premiers moments (moyenne et covariance) suffisent à déterminer complètement la loi de la densité de probabilité gaussienne relative à chaque classe.

### 4.2 ESTIMATION DU NOYAU DES CLASSES

a) Moyenne

Nous choisirons comme estimateur du centre des classes la moyenne arithmétique des données qui on été affectées à cette classe

$$\hat{\mu}_k(t_k) = \frac{1}{t} \sum_{i=1}^{t} x^i$$

$\hat{\mu}_k(t_k)$: estimateur du centre de la classe $C_k$ lorsqu'elle a absorbé t éléments

$t_k$: nombre d'observations affectées à la classe $C_k$

$x^i$: ième observation de la classe $C_k$

La formule récurrente permettant le calcul de la moyenne est

$$\hat{\mu}_k(t_k+1) = \hat{\mu}_k(t_k) + \frac{1}{t_k+1}\left[ x^{(t_k+1)} - \hat{\mu}_k(t_k)\right]$$

b) Matrice de covariance

L'estimateur de la matrice de covariance d'une classe $C_k$ sera la covariance empirique, calculée de façon récursive, à partir des éléments affectés à la classe $C_k$. Ici se pose naturellement le problème de l'initialisation puisque la "covariance empirique" n'a pas de sens lorsque la classe ne contient qu'un seul élément en début d'apprentissage. Nous reviendrons ultérieurement sur ce problème.

### 4.3 MESURE DE L'APPARTENANCE D'UN ÉLÉMENT À UNE CLASSE

Soit $P'(x/C_k)$ la densité de probabilité conditionnelle de la mesure (c'est-à-dire pour que l'on observe le vecteur x sachant que cette observation provient de la classe $C_k$), son expression est:

$$P'(c/C_k) = \frac{1}{(2\pi)^{\frac{N}{2}} \left| \det \sum_k \right|^{1/2}} e^{-1/2\left[(x-\mu_k)^T \sum_k^{-1} (x-\mu_k)\right]}$$

La mesure de l'appartenance d'un élément à une classe sera la probabilité à postériori $P(C_k/x)$ donnée par le théoreme de Bayes:

$$P'(C_k/x) = \frac{P'(x/C_k)/P(C_k)}{P'(x)}$$

où:
- $P'(C_k)$ est la densité de probabilité à priori d'obtenir la classe $C_k$
- $P'(x)$ est la densité de probabilité marginale de x.

L'algorithme va donc devoir calculer, pour tout $k \in \{1,2,\ldots,k\}$ (si K est le nombre de classes créées) la quantité $P'(C_k/x)$. Cependant deux remarques permettent de simplifier ce calcul:

i) P(x), qui est la densité de probabilité marginale de x ne dépend pas de la classe

ii) Nous pouvons faire l'hypothèse que toutes les classes ont la même probabilité à priori.

Dans ces conditions, le calcul de $P'(C_k/x)$ peut être remplacé par celui de $P'(x/C_k)$.

## 4.4 Règle de décision

Ayant calculé toutes les valeurs de $P'(x/C_k)$ il convient de choisir une règle permettant d'affecter la donnée x à une classe.

Cette règle sera celle du Maximum de Vraisemblance, autrement dit:

$$x \in C_{k^*} \quad \text{si} \quad P'(x/C_{k^*}) = \max_k [P'(x/C_k)] \geq S$$

$$k = 1,2,\ldots,K.$$

La donnée sera donc affectée à la classe dont elle est le plus proche à condition que cette proximité, traduite en termes de densité de probabilité soit supérieure à un certain seuil S.

- Si cette condition est remplie, l'estimation du centre et de la matrice de covariance de la classe $C_k$ sont alors modifiées pour tenir compte de ce nouvel élément,
- sinon, 2 possibilités peuvent se présenter:
  - ou bien, si le nombre maximum autorisé de classes n'est pas atteint, on créé une nouvelle classe à laquelle on affecte l'élément,
  - ou bien, si l'on ne peut plus créer de classes, l'élément est affecté à la classe $C_k$ mais sa moyenne et sa matrice de covariance ne sont pas modifiées. (Le programme signale dans ce cas:"élément classé sans apprentissage").

## 4.5 Initialisation

En ce qui concerne la moyenne, la classe est initialisée directement par la valeur du premier élément qui lui est affecté. Le problème est plus delicat en ce qui concerne la matrice de covariance.

En effet, en début d'apprentissage, lorsque la classe ne possède qu'un seul élément, la covariance empirique n'a pas de signification. Aussi faudra-t-il fournir au programme une matrice d'initialisation $\sum_o$, arbitraire, qui sera utilisée pour initialiser toutes les classes.

Ce choix de $\sum_o$ est en effet relativement délicat. Si l'on connaît au départ l'ensemble (ou une partie) des données on peut calculer, sur ces données la matrice de covariance. Ceci nous donnera une idée de leur dispersion et nous guidera dans notre choix de $\sum_o$. En tout cas on disposera au moins d'une borne supérieure pour $\sum_o$.

La transition entre la matrice d'initialisation $\sum_o$ et la covariance empirique se fera par exemple par la formule récurrente:

$$\sum_k (t_k+1) = \frac{1}{3} (2 \sum_k (t_k) + S_k (t_k+1))$$

où $S_k(t+1)$ est la matrice de covariance empirique de la classe k calculée sur $(t_k+1)$ éléments de la façon suivante:

$$S_k(t_k+1) = \frac{1}{t_k+1} \sum_{t_k=1}^{t_k+1} (x^t - \hat{\mu}(t_k))(x^t - \mu(t_k))^T$$

De cette façon on limite l'influence de la covariance empirique en début d'apprentissage puisqu'alors elle n'est pas très significative du fait du faible nombre d'éléments dans la classe.

## 4.6 Choix et rôle du seuil S

Le rôle du seuil S est le même que dans l'algorithme CLRO. Il nous donne l'assurance que l'affectation d'un élément à une classe se fait avec une probabilité supérieure à une certaine valeur S, fixée au départ.

La détermination du seuil S est relativement aisée ici, lorsque le choix de $\sum_o$ a été fait.

On choisira, en effet, comme valeur du seuil S, une fraction arbitraire de la densité de probabilité maximum relative à la classe vide, c'est-à-dire à la classe de covariance $\sum_o$; S est donc donnée par:

$$S = \frac{1}{(2\pi)^{\frac{N}{2}} \left| \det \sum_o \right|^{1/2}} \beta \; ; \qquad 0<\beta<1$$

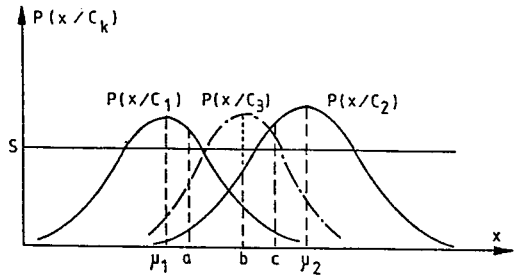



Fig. 13: Mécanisme de création de classes

Naturellement, plus la valeur de β se rapprochera de 1, plus le nombre de classes créées par l'algorithme aura tendance à augmenter.

## 4.7 INTERPRÉTATION GRAPHIQUE DU MÈCANISME DE CRÉATION DE CLASSES

On se placera dans le cas où 2 classes $C_1$ et $C_2$ ont été créées, en trait plein sur la figure 13

- si x=a $x \in C_1$ car $P'(x/C_1) > P'(x/C_2) > S$
- si x=c $x \in C_2$ car $P'(x/C_2) > P'(x/C_1) > S$
- si x=b $P'(x/C_2) > P'(x/C_1) < S$

donc création d'une nouvelle classe $C_3$ initialisée par:

$$\begin{cases} \mu_3 = b \\ \Sigma_0 \end{cases}$$

représentée en trait pointillé sur la figure 13.

## 4.8 VISUALISATION DES ELLIPOÏDES DE DISPERSION

Nous avons pensé qu'il serait intéresant de disposer à la fin de la classification d'une méthode permettant de visualiser les ellipsoïdes de dispersion relatives à chaque classe. Ces ellipsoïdes étant situées dans l'espace à N dimensions (N est le nombre de composantes de la donnée) il fallait pouvoir disposer de leur projection sur un plan défini par l'utilisateur. Ce plan sera naturellement défini par 2 axes relatifs à 2 composantes du "vecteur-donnée".

Notons qu'il s'agit bien d'ellipsoïdes puisque la densité de probabilité gaussienne multidimensionnelle donnée par:

$$f(x) = \frac{1}{(2\pi)^{n/2} \left| \det(\Sigma) \right|^{1/2}} e^{-1/2 \left[ (x-\mu)^T \Sigma^{-1} (x-\mu) \right]}$$

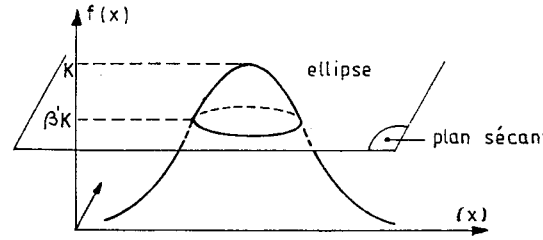
fait intervenir une forme quadratique en x; en la développant on obtient:

$$\left[ x_1-\mu_1 \; (x_2-\mu_2) \ldots (x_i-\mu_i) \ldots (x_n-\mu_n) \right] \begin{bmatrix} s_{11} & s_{12} \ldots s_{1j} \ldots s_{1n} & (x_1-\mu_1) \\ s_{21} & s_{22} \ldots s_{2j} \ldots s_{2n} & (x_2-\mu_2) \\ \vdots & & \\ s_{i1} & s_{i2} \ldots s_{ij} \ldots s_{in} & (x_i-\mu_i) \\ \vdots & & \\ s_{n1} & s_{n2} \ldots s_{nj} \ldots s_{nn} & (x_n-\mu_n) \end{bmatrix}$$

Les $S_{ij}$ étant les éléments de la matrice $\Sigma^{-1}$. En projetant dans le plan (i j) on obtient la forme quadratique suivante:

$$\left[ (x_i-\mu_i)(x_j-\mu_j) \right] \begin{bmatrix} S_{ii} & S_{ij} \\ S_{ji} & S_{jj} \end{bmatrix} \begin{bmatrix} (x_i-\mu_i) \\ (x_j-\mu_j) \end{bmatrix}$$

La densité de probabilité gaussienne de la forme $f(x) = K\, e^{-1/2 \left[ X^T Q x \right]}$ définit une hypersurface communément appelée "cloche de Gauss". Si on la coupe par un plan parallèle au plan (i,j) et défini par $f(x) = \beta' K$ --- $(0 < \beta' < 1)$ on obtiendra une ellipse d'equiprobabilité:




Eclipse d'équiprobabilité

on peut écrire:

$$\beta' K = K\, e^{-1/2 \left[ (X^T Q X) \right]}$$

$$\text{Log } \beta' = -\frac{1}{2} X^T Q T$$

soit $X^T Q T = -2 \text{ Log } \beta'$

et en explicitant la forme quadratique

$$\left[ (x_i-\mu_i)(x_j-\mu_j) \right] \begin{bmatrix} S_{ii} & S_{ij} \\ S_{ji} & S_{jj} \end{bmatrix} \begin{bmatrix} (x_i-\mu_i) \\ (x_j-\mu_j) \end{bmatrix} = -2 \text{Log } \beta'$$

en développant et en remplaçant les indices i et j par 1 et 2:

$$S_{11}x_1^2+S_{22}x_2^2+(S_{21}+S_{22})x_1x_2-(2S_{11}\mu_1+S_{21}\mu_2+S_{12}\mu_2)x_1-(2S_{22}\mu_2+S_{21}\mu_1+S_{12}\mu_1)x_2+(S_{21}+S_{22})\mu_1\mu_2+S_{11}\mu_1^2+S_{22}\mu_2^2\ 2\text{Log}\,\beta' = 0$$

équation que l'on indentifie directement --- avec l'équation générale d'une ellipse:

$$ax^2 + 2bxy + cy^2 + dx + cy + f = 0$$

Cette équation quadratique est difficile à - manipuler, aussi, moyennant cetains change-- ments de variables on se remènera à la forme classique de l'équation de l'ellipse par rap port aux axes principaux:

$$\frac{x^2}{u^2} + \frac{y^2}{v^2} - 1 = 0$$

que l'on écrira en coordonnées polaires:

$x = u \cos \theta$

$y = v \sin \theta$

## 5. APPLICATIONS

### 5.1 RECONNAISSANCE DES COMPOSANTS D'UN MÉ-- LANGE DE DENSITÉS DE PROBABILITÉS GAUS-- SIENNES BIDIMENSIONNELLES

L'exemple que nous proposons a été traité -- par d'autres mèthodes dans la thèse de A. -- Schroeder /9/ et de R. Lopez de Mantaras /7/.

Nous disposons de 150 points de $R^2$ tirés de 3 distributions gaussiennes bidimensionne--- lles de paramètres:

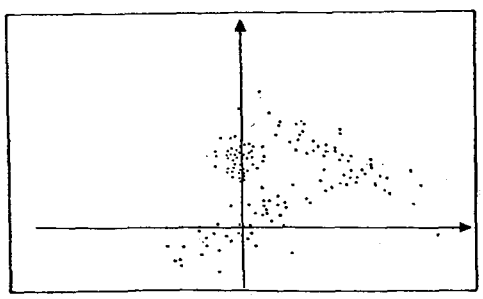


Fig. 14: Points à classer

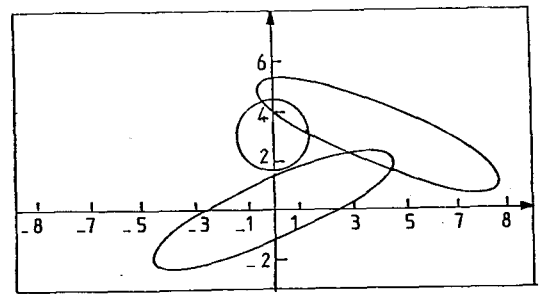



Fig. 15: Ellipses de dispersion réelles

$$x^1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad x^2 = \begin{bmatrix} 0 \\ 3 \end{bmatrix} \qquad x^3 = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$

$$S^1 = \begin{bmatrix} 4 & 1.7 \\ 1.7 & 1 \end{bmatrix} \quad S^2 = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.25 \end{bmatrix} \quad S^3 = \begin{bmatrix} 4 & -1.7 \\ -1.7 & 1 \end{bmatrix}$$

La figure 14 représente les 150 points dans le plan où les $X^i$ et les $S^i$ sont respectivement les moyennes et les matrices de cova--- riances des distributions.

La figure 15 reprèsente les ellipses de dispersion des 3 distributions.

L'algorithme CLGA a donné les résultats suivants:

1er cas: le nombre de classes autorisées -- est libre (égal à 10 en réalité). On constate que 4 classes sont créées. En effet si -- les classes 1 et 3 regroupent de façon presque exacte les distributions 1 et 3, la distribution 2 a été scindée en 2 classes au de meurant fort inégales puisque l'une ne com-- porte que 14 éléments.

La figure 16 montre les ellipses de disper-- sion des 4 classes obtenues. On donne également la "sortie" du programme avec l'estimation des moyennes et des matrices de cova--- riances. Tableau 1.

2ème cas: l'algorithme n'est autorisé qu'à créer 3 classes au maximum. Le résultat obtenu est alors presque parfait puisque ---- seuls deux éléments sur 150 sont mal clas-- sés. La figure 17 donne les ellipses de dis persion des classes tracées à partir des es timations de l'algorithme. On donne ensuite la "sortie" du programme. On peut comparer les estimations des moyennes et des matri-- ces de covariance avec les valeurs vraies.

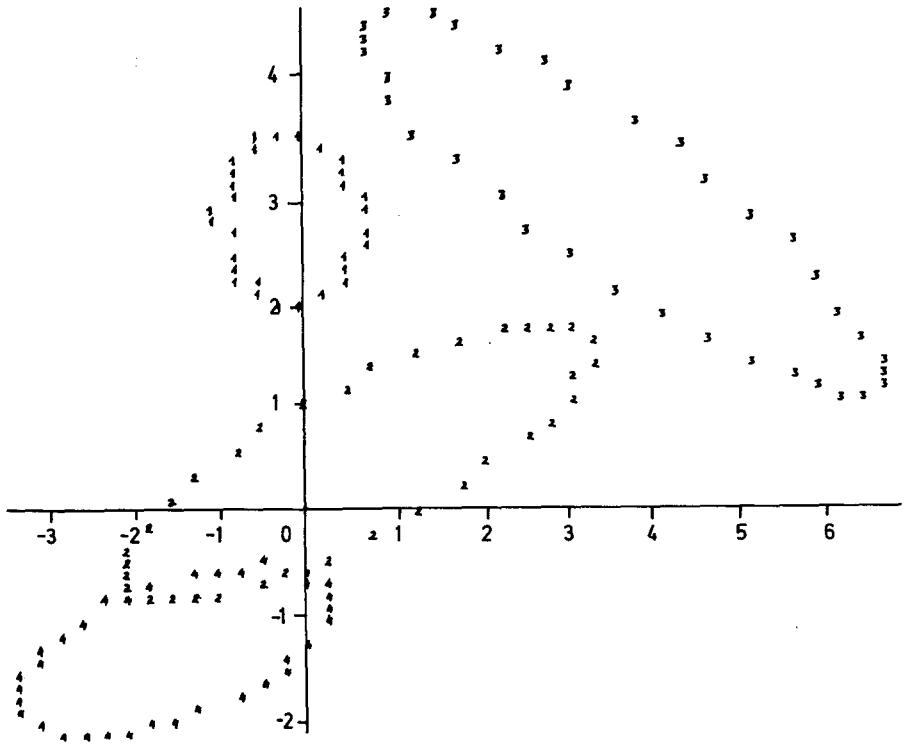


Fig. 16: Ellipses de dispersion obtenues avec l'algorithme

Tableau 1

Estimation des moyennes et des matrices de covariance des 4 classes obtenues

```
RESULTATS DE LA CLASSIFICATION:NOYAU DES CLASSES
CLASSE 1      VECTEUR DE MOYENNE
   -0.10          2.95
Σ1=[  0.219        -0.008 ]
   [ -0.008         0.228 ]
LA CLASSE  1CONTIENT  51ELEMENTS DONT:

  50ELEMENTS DE L ANCIENNE CLASSE 1
   0ELEMENTS DE L ANCIENNE CLASSE 2
   1ELEMENTS DE L ANCIENNE CLASSE 3
CLASSE 2      VECTEUR DE MOYENNE
    0.63          0.49
Σ2=[ 2.446         0.975 ]
   [ 0.975         0.606 ]
LA CLASSE  2CONTIENT  35ELEMENTS DONT:

   0ELEMENTS DE L ANCIENNE CLASSE 1
  35ELEMENTS DE L ANCIENNE CLASSE 2
   0ELEMENTS DE L ANCIENNE CLASSE 3
CLASSE 3      VECTEUR DE MOYENNE
    3.88          3.07
Σ3=[  3.002       -1.675 ]
   [ -1.675         1.181 ]
LA CLASSE  3CONTIENT  50ELEMENTS DONT:

   0ELEMENTS DE L ANCIENNE CLASSE 1
   1ELEMENTS DE L ANCIENNE CLASSE 2
  49ELEMENTS DE L ANCIENNE CLASSE 3
CLASSE 4      VECTEUR DE MOYENNE
   -1.63         -1.40
Σ4=[ 1.065         0.259 ]
   [ 0.259         0.215 ]
LA CLASSE  4CONTIENT  14ELEMENTS DONT:

   0ELEMENTS DE L ANCIENNE CLASSE 1
  14ELEMENTS DE L ANCIENNE CLASSE 2
   0ELEMENTS DE L ANCIENNE CLASSE 3
150.000
ICARF=  31ICARC=  4
        MATRICE DE CONTINGENCE
  0.3333    0.0       0.0067
  0.0       0.2333    0.0
  0.0       0.0067    0.3267
  0.0       0.0933    0.0
        DIFFERENCE ENTRE LES DEUX CLASSIFICATIONS= 0.0482
        DISTANCE ENTRE LES DEUX PARTITIONS= 0.1918
```

Tableau 2

Estimation des moyennes et des matrices de covariance

```
RESULTATS DE LA CLASSIFICATION:NOYAU DES CLASSES
CLASSE 1      VECTEUR DE MOYENNE
   -0.10          2.95
Σ1 = [  0.219       -0.008 ]
     [ -0.008        0.228 ]
LA CLASSE  1CONTIENT  51ELEMENTS DONT:

  50ELEMENTS DE L ANCIENNE CLASSE 1
   0ELEMENTS DE L ANCIENNE CLASSE 2
   1ELEMENTS DE L ANCIENNE CLASSE 3
CLASSE 2      VECTEUR DE MOYENNE
    0.24          0.17
Σ2 = [ 2.197        0.987 ]
     [ 0.987        0.662 ]
LA CLASSE  2CONTIENT  49ELEMENTS DONT:

   0ELEMENTS DE L ANCIENNE CLASSE 1
  49ELEMENTS DE L ANCIENNE CLASSE 2
   0ELEMENTS DE L ANCIENNE CLASSE 3
CLASSE 3      VECTEUR DE MOYENNE
    3,90          2,90
Σ3=[  1.716       -0.911 ]
   [ -0.911         0.737 ]
LA CLASSE  3CONTIENT  50ELEMENTS DONT:

   0ELEMENTS DE L ANCIENNE CLASSE 1
   1ELEMENTS DE L ANCIENNE CLASSE 2
  49ELEMENTS DE L ANCIENNE CLASSE 3
150.000
ICARF=  31ICARC=  3
        MATRICE DE CONTINGENCE
  0.3333    0.0       0.0067
  0.0       0.3267    0.0
  0.0       0.0067    0.3267
        DIFFERENCE ENTRE LES DEUX CLASSIFICATIONS= 0.0563
        DISTANCE ENTRE LES DEUX PARTITIONS= 0.0562
```

On constate que la meilleure estimation a - lieu pour la classe 1, ce qui n'est pas une surprise puisque c'est la moins "dispersée".

### 5.1.1 Application à des données bidimensionnelles non gaussiennes

Il s'agit des 127 points de $R^2$ représentés - dans la figure 11 et déjà classés par l'algorithme CLRO dans le paragraphe 3. On constate que CLGA fait dans ce cas un classement - identique à celui de CLRO. Les 5 groupes de points ont chacun constitué une classe (classés 1 à 5). Les points A et B placés aux sommets du carré constituant pour leur part une classe chacun (classes 6 et 7), les ellipses relatives aux 7 classes créées sont données dans la figure 18.

## 5.2 APPLICATION À LA RECONNAISSANCE D'OBJETS EN ROBOTIQUE

a) Description de l'expérience et du capteur:

Un capteur, dont nous décrivons très -- sommairement le principe, effectue un certain nombre de mesures sur des objets. (Une description détaillée du capteur se trouve - dans la thèse de M.Calzada /10/). Chaque objet sera donc caractérisé par un vecteur --- dont les composantes seront constituées par les mesures fournies par le capteur.

Le capteur utilisé est destiné en principe à l'asservissement automatique, en boucle - fermée, d'un bras manipulateur ou d'une machine outil travaillant à poste fixe. Le -- principe de la mesure consiste à balayer -- par un faisceau laser un plan de l'espace, appelé "plan de mesure" au moyen d'un miroir tournant; ce balayage est rendu parallèle - dans une zone coupant l'objet grâce à deux lentilles sphériques et on focalise le balayage parallèle dans la fenêtre d'une photocathode par deux lentilles.

Le signal électrique que donne la photocathode correspond au passage du faisceau --- dans la zone où se trouve l'objet, c'est un signal tout ou rien, selon que le faisceau tombe sur la fenêtre ou est caché par l'objet. C'est ainsi qu'à partir d'une mesure de temps on determine une distance avec la particularité que la mesure effectuée est - d'une grande précision (de l'ordre du dixième de millimètre .

La duplication de ce capteur nous donne les coordonnées des points d'occultation du --- faisceau par l'objet.

On fait varier le plan de mesure afin de -- pouvoir balayer toute la pièce, ce mouvement vertical est contrôlé par un moteur -- pas à pas.

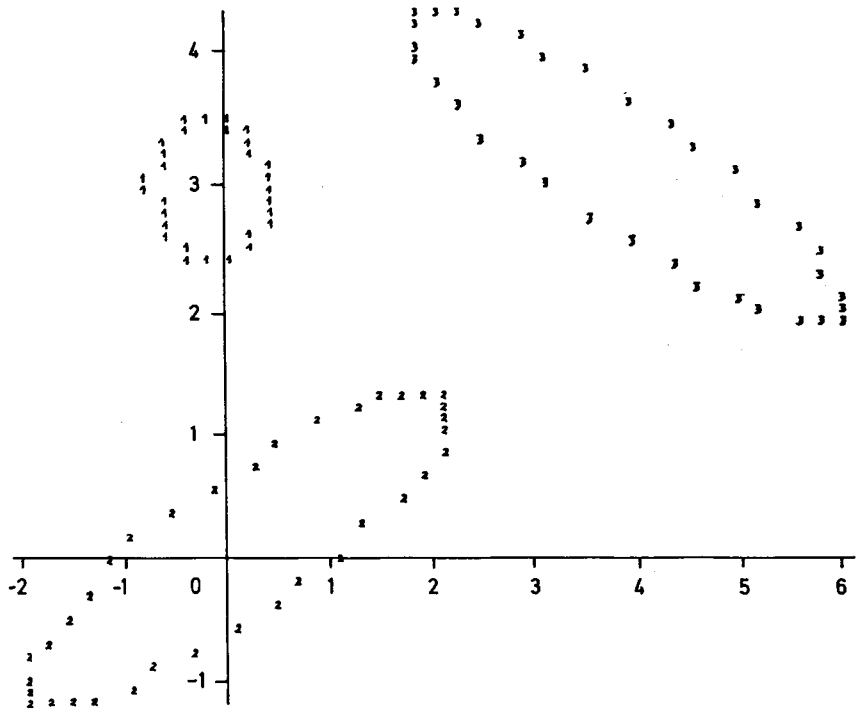



Fig. 17: Ellipses de dispersion obtenues avec 3 classes

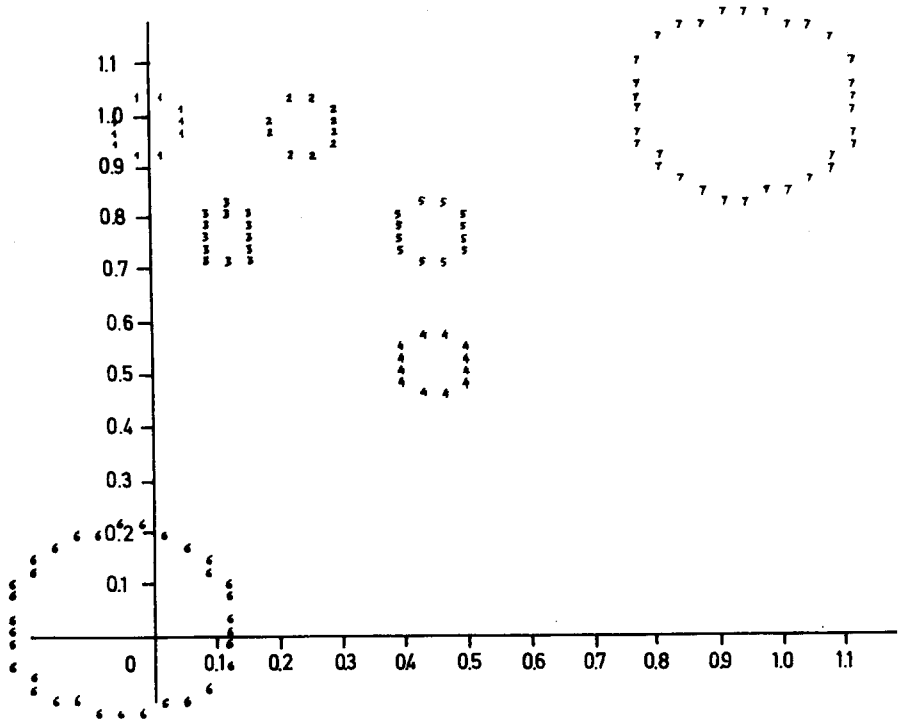



Fig. 18: Résultat de la classifitacion des points de la Fit. 11.

Dans l'expérience que nous décrivons par la suite, à chaque balayage d'un corps on n'a - tenu compte que des mesures correspondant à cinq plans. Le vecteur représentatif d'un objet à classer est composé de 2x10 éléments - selon le schéma de la figure 19. Chaque composante représente l'écart entre les points extrêmes de chacune des projections et le -- point milieu 0 de la troisième mesure, prise comme origine.

b) Application des algorithmes en mode professeur:

On utilisera les algorithmes précédem-- ment décrits (CLGA et CLRO pour faire de la reconnaissance d'objets. On imposera à cha-- que objet de reposer toujours sur la même face et sa reconnaissance devra se faire que-- lle que soit sa position angulaire. La phase d'apprentissage permettra de prendre en compte les différentes positions possibles de - l'objet. A cette fin on fait subir à l'objet une rotation de 180º par pas discrets, chacune de ces positions donnant lieu à une mesure par le capteur. L'algorithme sera utilisé en mode "professeur" c'est-à-dire que l'en-- semble des mesures ainsi effectuées pour un même objet sera forcé dans une même classe - qui sera donc caractéristique de l'objet considéré. Cependant, afin d'éviter une trop -- grande dispersion dans les mesures nous ---- avons été amené à créer trois classes par objet. Ainsi, partant d'une position initiale, une mesure relative à l'objet 0 sera affecté à la classe $0_1$ si la rotation subie est inférieure à $\pi/3$, à la classe $0_2$ si la rotation subie est comprise entre $\pi/3$ et $2\pi/3$ et à la classe $0_3$ si la rotation est comprise entre $2\pi/3$ et $\pi$.

Une fois créées ces classes de référence --- nous avons effectué une nouvelle série de mesures sur tous les objets et dans des posi--

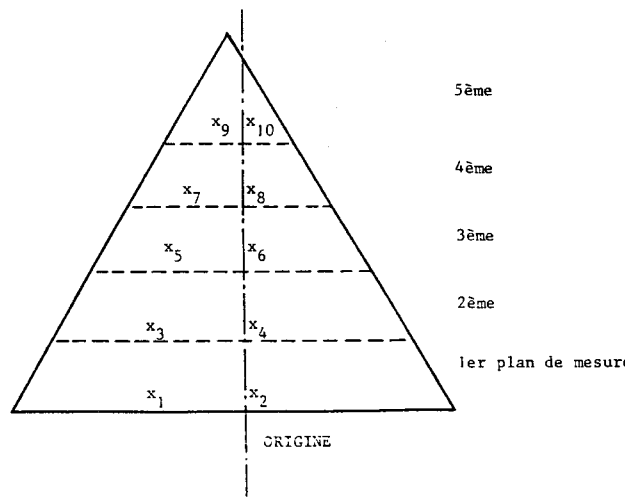



Fig. 9: Mesures faites sur l'objet à classer

tions quelconques afin de vérifier si l'algorithme les affectait bien dans leurs classes prévues.

Nous avons expérimenté cette méthode avec les 4 objets suivants: un cône, une pyramide, deux parallélépipèdes de sections différentes. Les objets on été choisis de dimensions très voisines, en particulier de même hauteur.

Les résultats obtenus on été les suivants:

- avec le programme CLGA: la reconnaissance s'est parfaitement effectuée pour tous les objets dans toutes les positions envisagées
- avec le programme CLRO: bien que dans l'ensemble la reconnaissance se soit effectuée correctement, on note cependant pour certaines positions particulières, une confusion entre le cône et la pyramide d'une part, et les 2 parallélépipèdes d'autre part. Pour plus de précision, on dira qu'il y a eu 5 "vecteurs-objets" mal classés sur un total de 69. Une solution pour pallier cet inconvénient serait de créer, lors de la phase d'apprentissage, un nombre de classes supérieur à 3 pour chaque objet, afin de réduire la dispersion des mesures dans une même classe.

Application des algorithmes en mode auto-apprentissage

Les données à classer ont été obtenues de la façon suivante: nous avons choisi 6 objets: un cylindre, 2 cônes de dimensions différentes, un tronc de cône, une pyramide et un cube dont la face supérieure présentait une concavité permettant la pose d'une sphère.

En réalité, on ne traitera pas dans cet exemple la reconnaissance de l'objet en entier mais d'une face de l'objet. Autrement dit, les mesures fournies par le capteur en X et Y seront séparées. La forme à classer sera donc une "face" de l'objet et sera représentée par un vecteur à 10 composantes (au lieu de 20 dans l'exemple précédent). Pour un même objet les mesures on été faites dans des positions sensiblement voisines. L'ensemble des données ainsi obtenues a été regroupé de telle manière que l'algorithme les traite dans un ordre aléatoire afin qu'il puisse créer librement des classes.

Il y avait au total 9 "faces" d'objets à reconnaître. En effet, certains objets, par suite de symétrie présentaient la même face en X et Y (cylindre, cônes, tronc de cônes). Le tronc de cône a été utilisé en reposant sur ces 2 faces. Le but de l'expérience était donc de vérifier que chaque classe créée correspondait bien à une "face" d'objet. Les résultats obtenus ont été les suivants:

- avec le programme CLGA: toutes les données relatives à un même objet ont été classées dans une même classe. La seule confusion observée l'a été entre certaines mesures relatives à la pyramide et

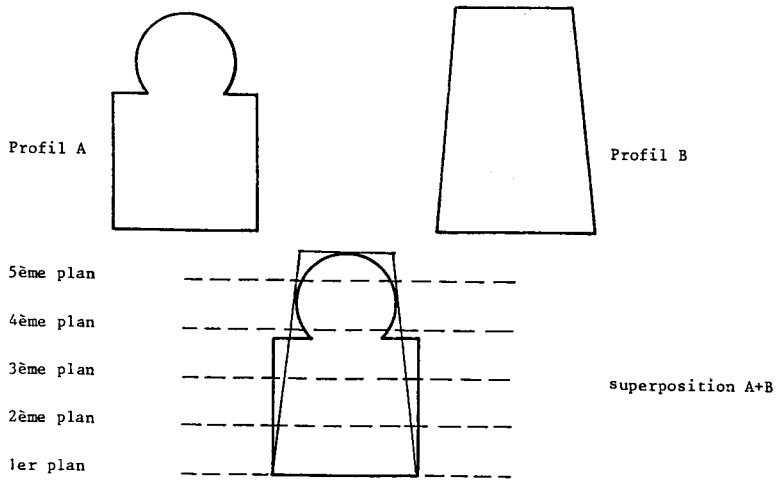



Figure 20

au cône de même hauteur. Cette "defai-- llance" s'explique aisément par le fait que lorsque la pyramide est positionnée de telle façon que les côtés de sa base soient parallèles aux axes de mesure le capteur la "voit" identique au cône.

- Avec le programme CLRO: ici une confu-- sion supplémentaire est observée entre certaines mesures relatives au tronc de cône d'une part et au cube surmonté de la sphère d'autre part. Dans tous les autres cas la classification s'est avérée correcte.

Pour éliminer cette confusion il aurait fa-- llu, pour chaque objet, faire un plus grand nombre de mesures. En effet, les 2 faces non différenciées par CLRO ont les profils que - montre la figure 20.

On constate alors que le profil des deux faces est assez proche, et qu'il aurait fallu plus de cinq plans de mesures pour bien les distinguer.

Conclusions

On constate d'abord que CLGA, bien que les données qu'il ait eu à traiter dans cette expérience n'aient aucune raison particulière d'être gaussiennes, donne de bons ré-- sultats. On peut donc en déduire qu'il est - assez "robuste" pour pouvoir traiter des données non gaussiennes dans de nombreus cas /4/.

Dans les deux expériences précédentes on note la supériorité de CLGA sur CLRO. Cepen--- dant cette supériorité s'exerce au prix ---- d'une plus grande complexité (place mémoire plus importante et surtout d'un temps cal-- cul beaucoup plus long, et ce d'autant plus que le nombre de composantes des vecteurs à classer est grand.

Le tableau suivant donne une idée de ces différences de temps pour classer 90 vecteurs - respectivement de 10 et 20 composantes.

| 90 vecteurs | CLGA | CLRO |
|---|---|---|
| 10 Composantes | 7 s | 0,64 s |
| 20 composantes | 1'20s | 1,2 s |

Ce tableau montre l'intérêt de CLRO dès que le nombre de composantes du vecteur devient important.

## 6. BIBLIOGRAPHIE


/1/ Diday, E. "La méthode des nuées dynamiques"; Revue de Statistiques Appli---- quées, vol. 19, nº 2, 1971.

/2/ Diday, E. et collaborateurs. "Optimisation en classification automatique" --- (2 tomes . Edité par l'Institut de Re-- cherche en Informatique et Automatique.

/3/ Aguilar Martin, J. "Learning and self learning procedures for automatic classification". Memorandum nº UCB/ERL M -- 78/51, Sept. 1978, Berkeley, Ca. USA.

/4/ Aguilar Martin, J.,Balssa, M., Calzada, M. "Perfomances comparées de deux algorithmes de classification avec apprentissage en robotique".

/5/ Aguilar Martin, J. "Algorithmes de classification intérative en l'absence d'information initiale". Cybernetica, nº 4, 1972.

/6/ Briot, M. "La stéréognosie en robotique application au tri de solides". Thèse - d'Etat, Université Paul Sabatier, Tou-- louse 1977.

/7/ Lopez de Mantaras Badia, R. "Auto-ap-- prentissage d'une partition; applica-- tion au classement itératif de données multidimensionnelles". Thèse 3ème Cycle Option Automatique, Université Paul Sabatier, Toulouse.

/8/ Aguilar Martin, J., Banon, G., Briot, - M., Lopez de Mantaras, R. "Tentative de simulation de l'agrégation et de classement des informations dans la reconaissance tactile de solides". Colloque BIO MECA II, Toulouse, Novembre 1976.

/9/ Schroeder, A. "Reconnaissance des composantes d'un mélange". Thèse de 3ème Cycle, Université Paris VI.

/10/ Calzada, M. "Procédé de détermination du posionnement précis d'une pièce uti-

lisant un capteur à faisceaux laser" Thèse 3ème Cycle, Université Paul Sabatier, Toulouse 1979.

/11/ Aguilar Martin, J., Briot, M. "Reconnaissance d'un solide à l'aide de l'estimation de la probabilité d'appartenance floue de son empreinte sur un capteur plan". Colloque International sur la Théorie et les Applications des Sous Ensembles Flous, Marseille, 20-22 Septembre 1978.

/12/ Balssa, M. "Analyse et conception de processus d'auto-apprentissage". Thèse de 3e cycle, Université Paul Sabatier, Toulouse, Octobre 1980.