

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique

Université Colonel El-Hadj Lakhdar. Batna
Faculté des Sciences de l'Ingénieur
Département Génie Industriel

MEMOIRE

Présenté au
Laboratoire d'Automatique et Productique

En vue de l'obtention du diplôme de

Magister

Spécialité: Génie Industriel

Option: Génie Industriel

Par

Rafik BENZAADI

Ingénieur d'état en Electronique,
Université de Batna

DIAGNOSTIC PAR RECONNAISSANCE DE FORMES - APPLICATION AUX SYSTEMES DE PRODUCTION

Directeur de Mémoire: **Dr. Hayet MOUSS**

Soutenu le: 03 décembre 2006, devant le Jury composé de:

Dr. N. MOUSS	<i>Président</i>	Université de Batna
Dr. H. MOUSS	<i>Rapporteur</i>	Université de Batna
Dr. A. DIB	<i>Examineur</i>	C. U. de Oum El-Bouaghi
Dr. A. LOUCHENE	<i>Examineur</i>	Université de Batna
Dr. F. ABDESSEMED	<i>Examineur</i>	Université de Batna
Dr. A. BETTA	<i>Examineur</i>	Université de Batna

Décembre 2006

SOMMAIRE

Chapitre 1: Introduction au contrôle et diagnostic des systèmes industriels

1.1. Introduction	1
1.1.1. Enjeux d'une entreprise moderne, et politiques de maintenance	1
1.1.2. Maintenance prédictive	1
1.1.3. Supervision et Diagnostic.....	2
1.2. Problème de diagnostic et commande tolérante aux fautes.....	3
1.3. Analyse de la propagation des défauts	6
1.4. Problème d'imprécision et limitation des modèles pour systèmes physiques complexes.....	6
1.5. Technique de Reconnaissance de Formes	6
1.6 Objectifs du mémoire	7
1.7 Organisation du mémoire	8

Chapitre 2: Reconnaissance de Formes Floue –concepts de bases

Introduction	9
2.1. Reconnaissance de formes.....	9
2.2. Théorie des sous-ensembles flous	11
– Opérateurs flous.....	13
2.3. Coalescence floue –algorithme des c -moyennes floues	13
2.4. Algorithme des c -Moyennes Strictes / Floues	14
2.5. Classification	16
– critère 1: règle des k -plus proches voisins: kppv (k -NN rule)	16
– critère 2: règle du plus proche prototype (NP rule)	16
– critère 3: règle du maximum.....	16

Chapitre 3: Approximateurs non linéaires à base des réseaux de neurones artificiels

3.1. Réseau de neurone.....	17
3.1.1. Application d'entrée (input mapping).....	18
3.1.2. Fonction d'activation	19
3.1.3. Le perceptron multicouche (MLP).....	20
3.1.4. Réseaux à fonctions radiales de bases (RBNN).....	21
3.2. Apprentissage –optimisation du gradient.....	22
3.2.1. Optimisation du gradient: une paire d'apprentissage.....	23
3.2.2. Optimisation du gradient: un ensemble de paires d'apprentissage	24
3.2.3. Discrétisation	25
3.3. Méthode des gradients conjugués	25

Chapitre 4: Diagnostic par reconnaissance de formes

4.1. Hypothèses, et analyse du problème	27
4.2. Approche proposée.....	27
4.2.1. Estimation des fonctions d'appartenance.....	27
4.2.2. Traiter une nouvelle observation.....	29
4.2.3. Détection et prévision de pannes	31
4.2.4. Diagnostic / décision.....	33
4.3. Conception et évaluation d'un système RdF.....	34

Chapitre 5: Application

5.1. Présentation du procédé.....	39
5.2. Modèle considéré.....	39
5.3. AMDEC.....	40
5.4. Simulation.....	41

Conclusion

Bibliographie

Chapitre 1

Introduction au Contrôle et Diagnostic des Systèmes Industriels

1.1. Introduction

1.1.1. Enjeux d'une entreprise moderne, et politiques de maintenance

Les équipements de fabrication modernes sont devenus plus gros, hautement complexes et fonctionnent avec un grand nombre de variables sous contrôle en boucle fermée. De plus la tendance de l'entreprise à améliorer le rapport qualité/prix dans un environnement caractérisé par une concurrence intense l'a poussé à chercher la réduction des coûts en aval, ceci a donné naissance à la fonction maintenance.

L'évolution des politiques de maintenance dépendait de l'évolution des technologies de production. Au début, la maintenance était essentiellement *corrective*, l'intervention de l'opérateur désigné n'est effectuée qu'en l'occurrence d'une panne soit pour réparer ou dépanner (réparer provisoirement pour assurer la continuité), la détection et diagnostic étaient basés sur la présence permanente de l'opérateur et son expérience. Cette politique ne pouvait pas être optimale, la machine pouvant être indisponible aux moments les plus critiques.

Cette fonction est ensuite évoluée vers la *maintenance préventive* qui consiste à intervenir sur l'équipement avant défaillance. La maintenance systématique est effectuée selon un échancier établi en fonction des durées de vie. La durée de vie d'un composant dans un équipement est estimée mathématiquement en fonction des constantes relatives au composant et ses conditions de fonctionnement. Bien que les temps d'arrêts sont planifiés et par suite les coûts contrôlés, cette politique n'est non plus optimale à cause des surcoûts de fonctionnement et arrêts non prévus (des pièces en bon état peuvent être remplacées ou des pièces endommagées avant durée prévue).

La *maintenance conditionnelle* a permis d'optimiser les temps d'arrêts et remplacement de pièces usées. Elle est basée sur l'analyse de grandeurs physiques traduisant l'état des éléments que l'on souhaite surveiller. Cependant une connaissance précise de ces éléments est nécessaire.

1.1.2. Maintenance prédictive

Les technologies avancées de traitement de l'information ainsi que la progression rapide des démarches de conception et optimisation des mathématiques de contrôle ont réorienté les méthodes de pilotage des opérations de maintenance dans les procédés industriels. En effet il y a disponibilité d'un grand nombre de données pour les calculateurs numériques en vue de traitement pour l'élaboration de décisions relatives aux applications de détection et diagnostic de défauts. La *maintenance prédictive* est une version avancée de la conditionnelle, elle est

basée sur l'observation et prédiction de la tendance de certaines grandeurs physiques vers la défaillance. Tous ces facteurs ont poussé à développer des algorithmes avancés de détection et diagnostic de pannes, basés essentiellement sur le contrôle adaptatif et techniques de l'intelligence artificielle.

1.1.3. Supervision et Diagnostic

Un système physique, ou procédé technologique, est conçu pour réaliser une tâche. Cette tâche peut être totalement ou partiellement remplie avec un certain niveau de qualité. Ces processus peuvent évoluer sous différents modes de fonctionnement, on peut citer:

- les modes normaux sous lesquels la mission est totalement remplie, l'un d'eux est le mode nominal correspondant à une qualité parfaite de la mission,
- les modes anormaux sous lesquels la mission est partiellement ou totalement non remplie, incluant les modes défaillants, les modes interdits, ...,
- les modes évolutifs qui caractérisent des modes, souvent transitoires, de passage d'un mode de fonctionnement à un autre.

Le diagnostic d'un système est basé sur l'identification du mode de fonctionnement, à chaque instant, par ses manifestations extérieures. Son principe général consiste à confronter les données relevées au cours du fonctionnement réel du système avec la connaissance que l'on a (même incomplète) sur son fonctionnement normal ou défaillant. Si le mode de fonctionnement identifié est un mode défaillant, le système de diagnostic doit localiser sa cause.

La supervision est ainsi définie comme étant le contrôle de l'exécution d'une opération ou d'un travail accompli par un homme ou par une machine. En présence de défaillances, la supervision prendra toutes les démarches nécessaires pour le retour du système vers un mode de fonctionnement normal. Un système de supervision traitant l'apparition d'un défaut peut être défini à travers les opérations suivantes:

- l'acquisition de mesures ou de comptes rendus permettant de vérifier que les consignes envoyées vers le processus produisent exactement les effets attendus, cela est réalisé par des capteurs,
- la perception qui fournit des données pré-traitées, par les techniques de traitement du signal, extraction de caractéristiques, ...
- le diagnostic qui consiste à déterminer dans quel état de fonctionnement se trouve le système. Il établit un lien de cause à effet entre un symptôme observé et la défaillance qui est survenue,

- éventuellement, le diagnostic prédictif qui permet une intervention préventive sur le système pour éviter qu'un mode non désiré ne se produise,
- enfin, la commande, c-à-d, décision: une séquence d'actions correctives destinées à rendre au système son état de fonctionnement normal.

Le diagnostic et la supervision continus d'un système industriel permettent d'améliorer la productivité par l'augmentation de la disponibilité.

1.2. Problème de diagnostic et commande tolérante aux fautes

Nous allons poser le problème de contrôle et diagnostic des systèmes industriels selon les points suivants:

- Le problème de commande et détection de défauts peut être formulé comme suit: Un système dynamique sous contrôle en boucle fermée se trouve en fonctionnement normal dans un environnement (ou configuration) E_1 . Cette configuration est changée vers E_2 à la suite d'un défaut qui survient à un moment t_0 . On suppose dans un premier lieu que le système ne peut se trouver que dans l'une des configurations de l'ensemble fini $\{E_i\}_{i=1,2,..,n}$ correspondant à une combinaison particulière dans l'espace des paramètres du système. Une configuration E_i est stable sous contrôle C_i , si toutes les configurations sont connues, il est simple de concevoir les commandes correspondantes $\{C_i\}$ avec un niveau de contrôle adaptatif permettant l'identification des paramètres du système sous défaillance et donc la mise à jour des paramètres du contrôleur afin de maintenir la réponse désirée.

Voici un exemple: la dynamique d'un hélicoptère dans un plan vertical, autour de conditions de fonctionnement typique (configuration E_1) est décrite par:

$$\mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B}_1 \mathbf{u}(k)$$

l'objectif est de commander ce système en présence des défaillances afin de maintenir la réponse désirée à la trajectoire de référence.

On suppose qu'après défaillance, le système peut se trouver dans l'une des configurations E_2 ou E_3 ; correspondant aux nouvelles matrices \mathbf{B}_2 et \mathbf{B}_3 . Dans ce cas, des matrices \mathbf{K}_1 , \mathbf{K}_2 et \mathbf{K}_3 d'une commande par retour d'état sont définis tels que:

- le procédé, en fonctionnement normal, est stable avec une réponse désirée quand la commande est commutée vers \mathbf{K}_1 ,
- les configurations E_2 et E_3 sont stables respectivement avec les matrices \mathbf{K}_2 et \mathbf{K}_3 ,
- les configurations E_2 et E_3 sont instables et différentes avec \mathbf{K}_1 .

Le même principe est appliqué si des coefficients estimés (une matrice \mathbf{G}) d'un observateur d'état sont employés, au lieu des chaînes de mesures, dans la boucle de retour.

Comme on l'a déjà mentionné, un mécanisme d'*identification* est nécessaire pour déterminer en temps réel la valeur actuelle de \mathbf{B}_i , c'est-à-dire le type de configuration, afin de *décider* la commande correspondante via une logique de commutation.

En général, ce même module peut être exploité pour le diagnostic. En effet, les coefficients du modèle sont fonctions de plusieurs paramètres (résistance, inductance, ...) du procédé, et par suite le changement d'une situation (l'état permanent et/ou certaines caractéristiques transitoires: temps de réponse, overshoot, ...), interprété par le changement de ce(s) coefficient(s) indique un changement au niveau de ces paramètres.

- On peut imaginer, pour un système de détection et diagnostic, l'architecture suivante:
 - ✚ Détection: présence (occurrence) de défaut,
 - ✚ Isolation: identifier l'élément responsable,
 - ✚ Estimation: amplitude, effets sur les autres parties du système, gravité

La commande tolérante aux fautes concerne une interaction entre un système donné (procédé) et un contrôleur. Le terme 'contrôleur' est employé ici pour indiquer un sens plus général: il inclut, en plus de la loi de commande, une couche de décision qui permet l'analyse du comportement afin d'identifier les défaillances et changer les paramètres de la loi de commande en sorte que le système en boucle fermée se trouve dans une région de performances acceptables.

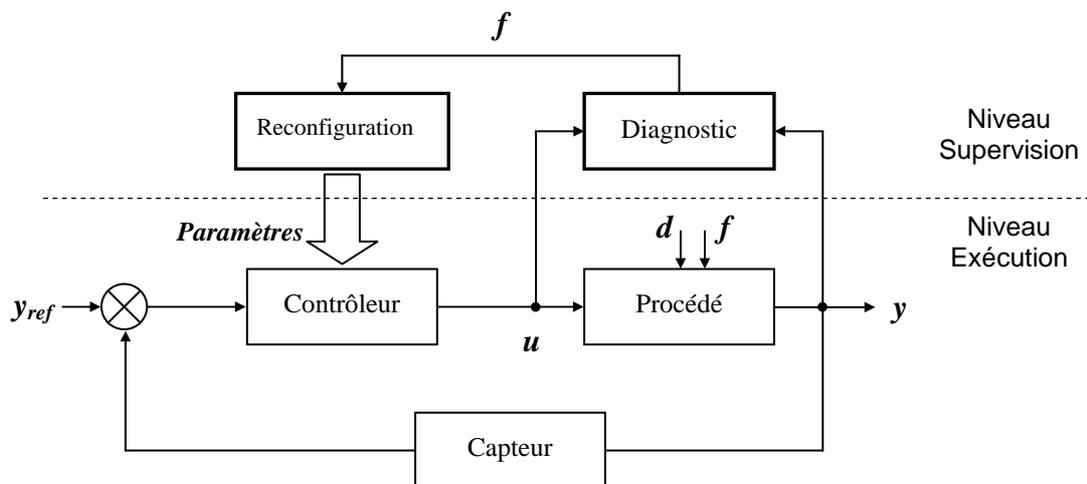


Figure 1.1: Commande Tolérante aux Fautes

Un système dynamique est pratiquement sujet de perturbations (d) et défauts (f). Le diagnostic consiste à identifier le défaut (f) étant donné une paire d'E/S (u, y).

Le procédé sujet de défaillance peut être décrit par:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B} \mathbf{u}(k) + \mathbf{E}_x \mathbf{d}(k) + \mathbf{F}_x \mathbf{f}(k) \\ \mathbf{y}(k) = \mathbf{C} \mathbf{x}(k) + \mathbf{D} \mathbf{u}(k) + \mathbf{E}_y \mathbf{d}(k) + \mathbf{F}_y \mathbf{f}(k) \end{cases}$$

Avec $\mathbf{x}(0) = \mathbf{x}_0$

\mathbf{x} le vecteur d'état,

\mathbf{u} le vecteur commande (entrées mesurées),

\mathbf{y} le vecteur sortie,

\mathbf{f} défauts à détecter,

\mathbf{d} les perturbations de nature aléatoire mais pas sujet de détection.

La détection et diagnostic consistent à concevoir un *générateur de résidus*, également un système décrit par:

$$\begin{cases} \mathbf{z}(k+1) = \mathbf{A}_z \mathbf{z}(k) + \mathbf{B}_{zu} \mathbf{u}(k) + \mathbf{B}_{z\mathbf{y}} \mathbf{y}(k) \\ \mathbf{r}(k) = \mathbf{C}_{rz} \mathbf{z}(k) + \mathbf{D}_{ru} \mathbf{u}(k) + \mathbf{D}_{ry} \mathbf{y}(k) \end{cases}$$

Avec $\mathbf{z}(0) = \mathbf{z}_0$,

En sorte que

- ✓ En l'absence de défauts, le résidu \mathbf{r} tend vers zéro quelque soit \mathbf{u} , \mathbf{d} , $\mathbf{x}(0)$ et $\mathbf{z}(0)$
- ✓ $\mathbf{r}(k)$ affecté par $\mathbf{f}(k)$ (non pas par \mathbf{u} ou d)

Un défaut est traduit par une modification au niveau des paramètres du procédé, c'est à dire les matrices \mathbf{A} , \mathbf{B} , \mathbf{C} et \mathbf{D} . A chaque valeur de résidu, correspond une combinaison de nouveaux paramètres que l'on peut identifier par des techniques appropriées.

Cette approche présente une solution pratique transitoire qui tolère la mise en marche du système de façon à ce que la tâche 'contrôle' ne se trouve pas suspendue. Mais jusqu'où ? (d) et (f) sont de nature aléatoire: un seuil d'alarme doit être précisé si la cadence et amplitude du défaut risquent d'atteindre un niveau inacceptable auquel le système ne peut plus être sous-contrôle avant que les éléments matériels responsables

ne soient changés ou réparés. De plus, on ne peut compter sur les caractéristiques statistiques pour calculer un remède à une situation catastrophique proche.

1.3. Analyse de la propagation des défauts

L'AMDEC (Analyse des Modes de Défaillances, de leurs Effets, et de leur Criticité) est une technique basée sur la description des modes des défaillances des constituants élémentaires et permet l'analyse d'effets importants causés par des éléments simples qui ne fonctionnent plus comme prévu. Une base de données avec des informations sur les modes de défaillances possibles (historiques des pannes, lois de probabilités, ..) est normalement disponible.

Elément	Mode de défaillances	Causes	Effets	gravité
...
...

Figure 1.2: éléments d'une analyse AMDEC

Une AMDEC permet d'avoir un aperçu général sur l'ensemble des effets possibles, c'est à dire, imaginer des régions de gravité différentes, donc des modes de fonctionnement/défaillances.

1.4. Problème d'imprécision et limitation des modèles analytiques pour systèmes physiques complexes

Il est très difficile de décrire, complètement et avec précision, un système physique réel par une équation différentielle à cause des non linéarités, du caractère stochastique des perturbations, et du fait que les paramètres caractéristiques changent dans le temps en fonction des conditions externes de fonctionnement.

Même les techniques de commande adaptatives qui prennent en charge ce dernier problème risquent de n'être valables que sur des intervalles donnés à cause de l'architecture fixe du modèle supposé (nombre des E/S, nombre d'états, nombre de paramètres).

1.5. Technique de la Reconnaissance de Formes

Aujourd'hui, la complexité des procédés technologiques a compliqué les mathématiques de contrôle et a donné naissance à de nouvelles méthodes inspirées du mécanisme de raisonnement humain: supposer l'existence d'états de fonctionnements, pouvoir traiter des classes de mesures similaires au lieu/en plus de données numériques précises (insister sur la précision là où il faut, mais après avoir eu une image globale sur la façon de faire le raisonnement), mettre en évidence l'apparition de nouvelles situations, définir et ajuster

l'architecture et paramètres de contrôle, reconnaître une situation, un changement, prévoir une future situation, ..etc. Tels sont les aspects de la Reconnaissance de Formes (RdF).

La RdF se base sur la classification des états observés en les comparant à des formes types. Une forme caractérise un état typique et regroupe l'ensemble des états similaires qui ne sont pas forcément les mêmes. L'observation du fonctionnement du processus est représentée par des points dans un espace à n dimensions appelé l'espace de représentation. Les formes types indiquent les modes de fonctionnement représentés par des ensembles de points occupant des zones restreintes de l'espace de représentation, et sont appelées classes.

L'application de la RdF à la détection et diagnostic se heurte souvent aux problèmes suivants :

- dans une base de connaissance incomplète, les modes de fonctionnement ne sont pas tous représentés. En conséquence, un module de diagnostic doit être adaptatif afin d'inclure à sa base de connaissance les nouveaux modes dès qu'ils apparaissent,
- lorsque le système évolue vers un mode anormal, il est nécessaire d'anticiper cette évolution plutôt que d'attendre d'arriver à ce mode afin d'éviter ses conséquences surtout s'il est dangereux. En conséquence le module de diagnostic doit être prédictif,
- dans le cas d'un système évolutif, la base de connaissance doit être enrichie grâce à l'information apportée par les nouvelles observations. Cet enrichissement doit être réalisé en temps réel.

1.6. Objectifs du mémoire

Ce travail est réalisé au sein de l'équipe *Systèmes Sûrs et Performants* du LAP (Laboratoire d'Automatique et Productique) de l'Université de Batna. Parmi les objectifs de cette équipe la mise au point de modules de diagnostic et supervision de processus industriels en utilisant les techniques intelligentes. Dans ce cadre, on s'intéresse à la conception d'un système (logiciel-matériel) qui prend en charge des tâches de nature humaine, et l'appliquer au problème de détection et diagnostic de pannes, dans un procédé industriel complexe, en se basant sur la technique de la Reconnaissance de Formes. En effet, la supervision des processus industriels permet d'accroître la productivité et de diminuer le coût d'entretien par l'amélioration de la disponibilité des systèmes de production. Ces processus sont susceptibles d'évoluer, sous deux types de modes de fonctionnement: des modes normaux et des modes anormaux. Un mode de fonctionnement anormal correspond à un dysfonctionnement du système entraînant une réalisation partiellement ou complètement non conforme de la tâche pour laquelle ces processus sont conçus.

Le diagnostic est une composante principale d'un module de supervision. Il consiste à déterminer à chaque instant le mode de fonctionnement dans lequel le système se trouve et

identifier les causes dans le cas d'anomalies. Il s'appuie sur une connaissance à priori d'un nombre minimum des modes de fonctionnements déjà connus, et sur une connaissance instantanée matérialisée par une nouvelle observation de l'état du système. Il existe plusieurs approches pour réaliser le diagnostic, le choix d'une approche est lié au mode de représentation de la connaissance. En se basant sur l'hypothèses de connaissances représentables, donc traitables, numériquement, et d'un modèle analytique du procédé, complètement ou partiellement inconnu, l'approche par Reconnaissance des Formes (RdF) est particulièrement adaptée.

1.7 Organisation du mémoire

Le présent travail est une contribution qui se situe en ligne des travaux menés jusqu'ici sur la supervision des procédés industriels, utilisant des outils de l'intelligence artificielle. Ainsi et à travers les points énumérés et les avancées du problème, notre mémoire est organisé en cinq chapitres. Le premier chapitre se veut une présentation de généralités sur le diagnostic des systèmes industriels ainsi qu'un aperçu sur les méthodes à base de modèles. Le chapitre deux présente les concepts de base de la reconnaissance des formes et les fondements de la méthode adoptée. En chapitre trois, nous avons présenté un des éléments de l'IA, les réseaux de neurones artificiels, en tant que approximateurs de fonctions d'appartenance, en effet l'information sur les degrés d'appartenance est essentielle pour élaborer une décision de classification en temps réel. Le chapitre quatre est réservé essentiellement à l'approche proposée, une méthode de diagnostic, basée sur la technique de Reconnaissance de Formes, répondant aux objectifs précités. L'évaluation de la validité de la méthode est explicitée au chapitre cinq. Dans ce contexte, nous appliquons le système de diagnostic conçu sur un procédé chimique simple. Nous considérons ainsi un modèle analytique présenté dans la littérature juste pour générer des séquences d'état (créer des anomalies) et tester l'efficacité de l'approche de détection et aide à la décision.

Chapitre 2

Reconnaissance de Formes Floue – Concepts de base

Introduction

D'autres variables, inconnues ou non directement mesurables, doivent faire partie du vecteur d'état pour mieux décrire le comportement du système: La précision du modèle, objectif très difficile, est nécessaire pour mieux gérer les informations incertaines et imprévues. Cependant, l'expert humain montre des compétences supérieures dans la commande de machines et localisation de défauts, et peut faire un diagnostic correct bien qu'il ne dispose d'aucun modèle sur le système sous contrôle, en effet, le cerveau humain est apte à apprendre, gérer des informations imprécises et agir en fonction d'une combinaison complexe de signaux de capteurs au lieu de sources séparées d'informations.

2.1. Reconnaissance de Formes

La reconnaissance de forme (RdF), branche de l'Intelligence Artificielle, s'intéresse au problème d'identification de situations 'significatives' d'un système dans un environnement complexe ou noyé dans le bruit.

La RdF contribue aux approches de contrôle robuste de systèmes complexes, l'opérateur humain chargé de surveiller un procédé de fabrication décide parfois des commandes sur la base de 'classes d'informations' au lieu de valeurs précises pour certains paramètres. De plus, la nature et caractéristiques de ces classes sont 'optimisées' dans le temps, durant son expérience. Ceci nous conduit à déduire deux aspects relatifs à l'intelligence: raisonnement flou ou incertain, et apprentissage. La technique RdF permet d'implanter ces facultés en productique afin de résoudre des problèmes complexes tels que diagnostic.

Nous présentons ci-après des indications générales sur la conception d'un système de reconnaissances de formes floue.

- Structures de données

Le principe de la reconnaissance de formes repose sur la représentation d'un objet (figure 2.1) par un vecteur de q paramètres appelé *vecteur forme*, soit par un point de l'espace \mathcal{R}^q , appelé *espace de représentation*. Un ensemble d'objets $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, est donc représenté par un nuage de points.

L'objet k désigne une entité physique (personne, avion, enregistrement sismique, image, etc.), et est décrit par le vecteur forme: $\mathbf{x}_k = [x_{k1}, \dots, x_{kq}]^T$ avec x_{kj} la $j^{\text{ème}}$ caractéristique liée à l'objet k (température, pression, poids, ..etc).

Un autre type de structure consiste à exprimer des mesures de similarité (ex. distance) au lieu d'ensemble d'objets, c'est à dire, considérer une matrice de la forme:

$$\mathbf{R} = [r_{jk}] = [\rho(o_j, o_k)],$$

Où r_{jk} est une mesure de similarité par laquelle les objets o_j et o_k sont liés selon une certaine relation binaire ρ .

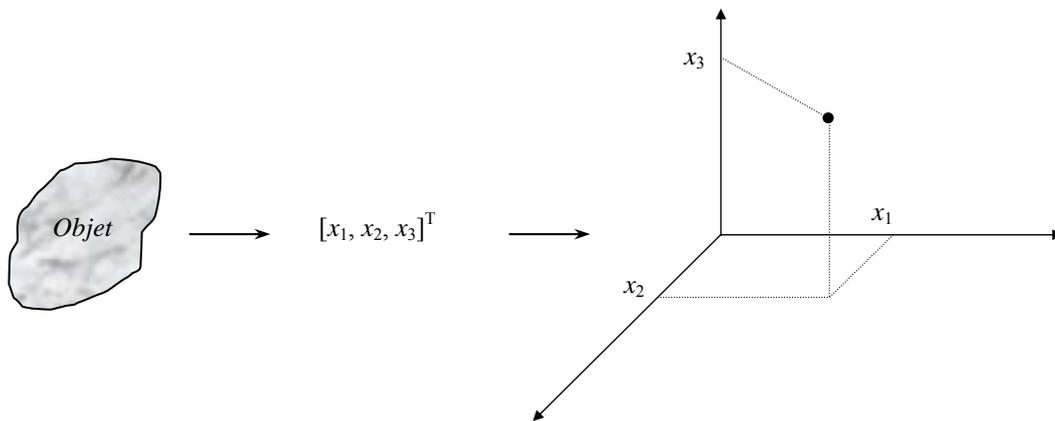


Figure 2.1: Représentation d'objets en RdF

- Les axes de l'espace de représentation sont choisis de sorte que la condensation ou dispersion du nuage de points (objets de l'ensemble \mathbf{X}) indique d'une certaine façon l'existence de *formes* typiques distinctes. En supposant qu'il existe dans l'ensemble d'apprentissage \mathbf{X} , c formes-type que l'on souhaite distinguer (figure 2.2), les éléments considérés comme représentatifs de ces formes-type sont appelés *prototypes*. Dans un cas réel, les nouvelles observations sont bruitées, un nouvel objet sera rarement identique à l'un des prototypes, on introduit alors la notion de *classes* notées $\omega_1, \omega_2, \dots, \omega_c$ qui correspondent à des zones de l'espace \mathfrak{R}^q , représentant les formes semblables.
- On distingue deux phases dans l'élaboration d'un processus de reconnaissance de formes:
 - 1) Une phase d'analyse et d'apprentissage dont l'objectif est de construire une fonction de décision $D: \mathfrak{R}^q \rightarrow \mathfrak{R}^c$

$$\mathbf{x}_i \mapsto \omega_j$$

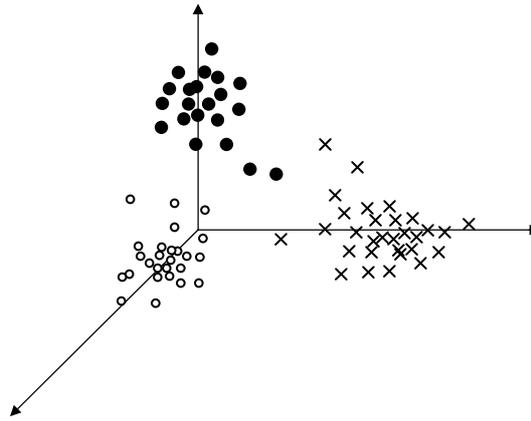


Figure 2.2: La dispersion du nuage de points indique l'existence de formes typiques

- ✓ Définition de l'espace de représentation: identification et sélection des paramètres décrivant les objets $\{\mathbf{x}_i\}$. Les performances du classifieur dépendent du choix du vecteur forme, des techniques telles que l'ACP (Analyse en Composantes Principales) aident à la résolution du problème.
 - ✓ Recherche d'une structure des observations en classes: technique de coalescence floue (apprentissage non supervisé).
 - ✓ Définition de la règle de décision établissant les frontières entre les différentes classes (conception du classifieur).
- 2) Une phase d'exploitation et d'optimisation:
- ✓ Définir à quelle classe appartient une nouvelle observation \mathbf{x}_{n+1} ,
 - ✓ Utiliser les nouvelles observations pour optimiser les performances du classifieur.

Nous allons voir dans les paragraphes qui suivent, l'apport de la logique floue au problème de structuration en classes dans l'approche RdF.

2.2. Théorie des sous-ensembles flous

Le cerveau humain est capable d'intégrer la notion d'imprécision de grandeurs non qualitatives dans son raisonnement, et cette faculté le rend plus performant qu'un ordinateur pour la réalisation de tâches imprécises par nature, comme par exemple, déchiffrer une écriture mal formée, reconnaître un visage qui a changé, identifier la signification d'un bruit anormal, ...etc.

Quand la complexité d'un système augmente, l'aptitude de faire un raisonnement correct sur le comportement diminue vers un seuil au-dessous duquel 'précision' et 'signification' deviennent des caractéristiques exclusives (principe d'incompatibilité). Le facteur-clé dans le raisonnement humain n'est pas des nombres, mais *étiquettes* de sous-ensembles flous, c'est à dire des classes d'objets pour lesquelles le changement entre état 'd'appartenance' et 'non appartenance' sera graduel.

Dans la théorie des ensembles classiques, une observation \mathbf{x} appartient ou n'appartient pas à un ensemble donné \mathbf{H} , et ne peut appartenir qu'à un ensemble unique. Cette appartenance se traduit mathématiquement par la fonction:

$$m_{\mathbf{H}}(\mathbf{x}) = \begin{cases} 1; & \text{si } \mathbf{x} \in \mathbf{H} \\ 0; & \text{si } \mathbf{x} \notin \mathbf{H} \end{cases}$$

Exemple 1:

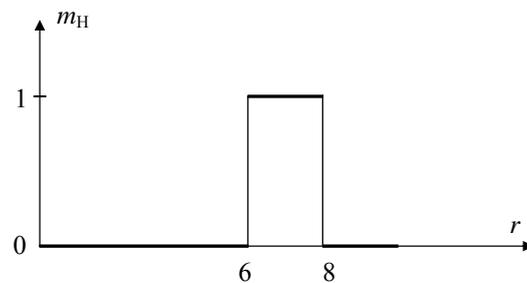
$$\mathbf{H} = \{r \in \mathfrak{R}; 6 \leq r \leq 8\}$$

$$m_{\mathbf{H}}: \mathfrak{R} \rightarrow \{0,1\}$$

$$m_{\mathbf{H}}(r) = \begin{cases} 1; & \text{si } 6 \leq r \leq 8 \\ 0; & \text{si ailleurs} \end{cases}$$

$$m_{\mathbf{H}}(5) = m_{\mathbf{H}}(9) = 0$$

$$m_{\mathbf{H}}(7.2) = 1$$



La théorie des sous-ensembles flous est introduite par L. A. Zadeh en 1965 pour décrire des variables linguistiques ambiguës, et réellement employées dans le raisonnement humain, tels que, *grand, très grand, petit, haut, ...*

L'appartenance d'une observation \mathbf{x} à un ensemble flou \mathbf{F} est graduelle, et donnée par une *fonction d'appartenance* $m_{\mathbf{F}}(\mathbf{x})$ qui prend ses valeurs dans l'intervalle $[0,1]$ au lieu de l'ensemble $\{0,1\}$. $m_{\mathbf{F}}(\mathbf{x})$ est dit *degré d'appartenance* de l'observation \mathbf{x} à l'ensemble \mathbf{F} .

Un sous-ensemble \mathbf{F} de \mathfrak{R}^q est un ensemble de couples $\{\mathbf{x}, m_{\mathbf{F}}(\mathbf{x})\}; \forall \mathbf{x} \in \mathfrak{R}^q$.

Cette définition montre que chaque objet \mathbf{x} de l'espace de représentation \mathfrak{R}^q appartient à toutes les classes $\{\omega_j\}$! mais à un degré d'appartenance, noté $\mu_{\omega_j}(\mathbf{x})$.

Exemple 2:

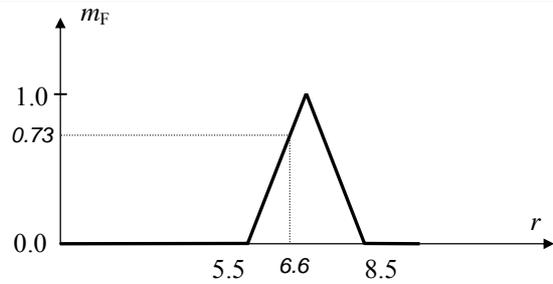
Considérons le sous ensemble flou: $\mathbf{F} = \{r \in \mathfrak{R}; r \text{ est proche de } 7\}$,

L'expression 'proche de' est une propriété floue qu'on peut décrire –par exemple– par la fonction $m_{\mathbf{F}}(r)$ qui a les propriétés suivantes:

- normalité: $m_F(7) = 1$
- monotonie: $(r \rightarrow 7) \Rightarrow (m_F(7) \rightarrow 1)$
- symétrie: $m_F(7-a) = m_F(7+a)$

Remarquons que 6.6 est considéré *proche de 7* au degré 0.73, et que tout réel en dehors de

$[5.5, 8.5]$ est absolument non proche de 7. En effet, 6.6 peut aussi être considéré comme *non proche de 7*, mais au degré 0.27!



– Opérateurs flous

Soient **A** et **B** deux sous-ensembles flous de l'ensemble **X**, et **x** un élément quelconque de **X**. les opérations élémentaires de logique floue sont définies comme suit:

- égalité (=): $(\mathbf{A} = \mathbf{B}) \Leftrightarrow (m_{\mathbf{A}}(\mathbf{x}) = m_{\mathbf{B}}(\mathbf{x}))$
- inclusion (\subset): $(\mathbf{A} \subset \mathbf{B}) \Leftrightarrow (m_{\mathbf{A}}(\mathbf{x}) \leq m_{\mathbf{B}}(\mathbf{x}))$
- complémentarité (\neg): $m_{\neg \mathbf{A}}(\mathbf{x}) = 1 - m_{\mathbf{A}}(\mathbf{x})$
- intersection (\cap ou AND logique flou): $m_{\mathbf{A} \cap \mathbf{B}}(\mathbf{x}) = \min\{m_{\mathbf{A}}(\mathbf{x}), m_{\mathbf{B}}(\mathbf{x})\}$
- union (\cup ou OR logique flou): $m_{\mathbf{A} \cup \mathbf{B}}(\mathbf{x}) = \max\{m_{\mathbf{A}}(\mathbf{x}), m_{\mathbf{B}}(\mathbf{x})\}$

Exemple 3:

$\mathbf{P} = \{\text{personnes}\}; x = h(p) = \text{taille de } p \in \mathbf{P}$

$\mathbf{X} = h(\mathbf{P}) = \{\text{tailles de } p \in \mathbf{P}\} = [0, 11]$

On définit les deux classes de tailles suivantes:

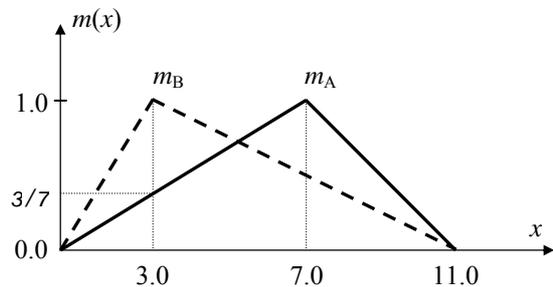
$\mathbf{A} = \{\text{personnes de taille proche de 7 pieds}\}$

$\mathbf{B} = \{\text{personnes de taille proche de 3 pieds}\}$

$m_{\mathbf{A}}(3) = 3/7$: degré par lequel une personne de taille 3 pieds est considérée de taille 7 pieds!

$m_{\neg \mathbf{A}}(3) = 1 - 3/7 = 4/7$: degré par lequel un 3 pieds n'est pas un 7 pieds!

$m_{\mathbf{A} \cap \mathbf{B}}(6) = \min\{6/7, 5/8\} = 5/8$: degré par lequel un 6 pieds est à la fois un (3 et 7) pieds!



Une question importante se pose à ce niveau: comment, et dans quelles conditions, est définie une fonction d'appartenance? ou, comment est évalué un degré d'appartenance?

2.3. Coalescence floue– algorithme des c-moyennes floues

Etant donné l'ensemble d'apprentissage $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, Le problème de coalescence dans \mathbf{X} consiste à attribuer aux objets $\{\mathbf{x}_k\}$ des coefficients qui indiquent une partition naturelle en sous-groupes de \mathbf{X} . L'objectif est donc de rechercher une structure en c classes, intuitivement séparables, dont l'appartenance est quantifiée par les *degrés d'appartenance*.

Avant de présenter le fameux algorithme des c -moyennes floues, commençons par expliquer l'idée de coalescence. On définit les ensembles de matrices d'étiquettes suivants:

$$\mathbf{M}_{fcnu} = \{ \mathbf{U} \in \mathbb{R}^{cn} \mid u_{ik} \in [0,1] \forall i, k \text{ et } 0 < \sum_{k=1}^n u_{ik} < n \}, \text{ (unconstrained fuzzy labels),}$$

$$\mathbf{M}_{fcn} = \{ \mathbf{U} \in \mathbf{M}_{fcnu} \mid \sum_{i=1}^c u_{ik} = 1 \forall k \}, \text{ (constrained fuzzy labels),}$$

$$\mathbf{M}_{cn} = \{ \mathbf{U} \in \mathbf{M}_{fcn} \mid u_{ik} \in \{0,1\} \forall i, k \}, \text{ (hard labels),}$$

c et n sont des indices qui indiquent respectivement: nombre de classes, et nombre d'échantillons de l'ensemble d'apprentissage.

Un *algorithme de coalescence floue* (ou *stricte*) est une application:

$$A: \mathbf{X} \rightarrow \mathbf{M}_{fcnu} \text{ (} \mathbf{M}_{fcn} \text{ ou } \mathbf{M}_{cn} \text{)}$$

$$\{ \mathbf{x}_i \} \mapsto \mathbf{U} = [u_{ik}]$$

où $\mathbf{x}_k = [x_{k1}, \dots, x_{kq}]^T$: le vecteur forme, et u_{ik} : degré d'appartenance de l'objet k à la classe i .

Le tableau 2.1 montre un exemple de partition typique en deux classes de l'ensemble $\mathbf{X} = \{x_1, x_2, x_3\} = \{\text{pêche, prune, nectarine}\}$.

Tableau 2.1: Exemple de partition stricte et floue.

	<i>Partition stricte</i> $\mathbf{U}_1 \in \mathbf{M}_{23}$	<i>Partition floue</i> $\mathbf{U}_2 \in \mathbf{M}_{f23}$	<i>Partition floue</i> $\mathbf{U}_3 \in \mathbf{M}_{f23u}$
Classes	$x_1 \quad x_2 \quad x_3$	$x_1 \quad x_2 \quad x_3$	$x_1 \quad x_2 \quad x_3$
<i>Pêche</i>	$\begin{bmatrix} 1.0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0.9 & 0.2 & 0.4 \end{bmatrix}$	$\begin{bmatrix} 0.9 & 0.5 & 0.5 \end{bmatrix}$
<i>Prune</i>	$\begin{bmatrix} 0 & 1.0 & 1.0 \end{bmatrix}$	$\begin{bmatrix} 0.1 & 0.8 & 0.6 \end{bmatrix}$	$\begin{bmatrix} 0.6 & 0.8 & 0.7 \end{bmatrix}$

2.4. Algorithme des c -Moyennes Strictes / Floues (Hard / Fuzzy c -Means) ou HCM / FCM

Cet algorithme est basé sur la minimisation d'un critère du type 'moindres carrés':

$$J_m(\mathbf{U}, \mathbf{v} : \mathbf{x}) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^m D_{ikA}^2 \quad (2.1)$$

où, c : nombre de classes,
 n : nombre d'objets dans l'ensemble d'apprentissage,
 \mathbf{U} : matrice de partition floue $\in \mathbf{M}_{fcn}$,

$m \in [1, \infty)$: degré flou de la partition, égal à 1 pour une coalescence stricte,

$\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_c\}$: vecteurs prototypes,

$$D_{ikA}^2 = \|\mathbf{x}_k - \mathbf{v}_i\|_A^2 = (\mathbf{x}_k - \mathbf{v}_i)^T \mathbf{A} (\mathbf{x}_k - \mathbf{v}_i),$$

\mathbf{A} : une matrice définie positive,

$\mathbf{I} = \{1, \dots, c\}$: Indices de classes.

L'optimisation de $J_m(\mathbf{U}, \mathbf{v}; \mathbf{x})$ conduit à l'algorithme HCM / FCM:

Entrées: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, c , m , $\|\cdot\|_A$, $\varepsilon > 0$,

Initialisation: - choix aléatoire de \mathbf{U} ,

$$\text{- Calcul des prototypes: } \mathbf{v}_{i,0} = \frac{\sum_{k=1}^n u_{ik}^m \mathbf{x}_k}{\sum_{k=1}^n u_{ik}^m} \text{ pour } i = 1, 2, \dots, c \quad (2.2)$$

Faire (à l'itération t)

1. degrés d'appartenance: soit $\mathbf{I}_t = \{i \in \mathbf{I} \mid D_{ik} = 0; 1 \leq i \leq c\}$; $k = 1, 2, \dots, n$
 ✓ pour $m > 1$

$$\text{Si } \mathbf{I}_t = \emptyset, \text{ alors } u_{ik} = \left[\sum_{j=1}^c \left(\frac{\|\mathbf{x}_k - \mathbf{v}_i\|_A}{\|\mathbf{x}_k - \mathbf{v}_j\|_A} \right)^{\frac{2}{m-1}} \right]^{-1} \text{ pour } i = 1, 2, \dots, c$$

Si $\mathbf{I}_t \neq \emptyset$, alors $u_{ik} = 0 \forall i \in (\mathbf{I} - \mathbf{I}_t)$, et arbitrairement $\sum_{s \in \mathbf{I}_t} u_{sk} = 1$

✓ pour $m = 1$,

$$u_{ik} = \begin{cases} 1; & \|\mathbf{x}_k - \mathbf{v}_i\|_A < \|\mathbf{x}_k - \mathbf{v}_j\|_A \\ 0 & \text{ailleurs} \end{cases}$$

2. mise à jour des prototypes $\{\mathbf{v}_{i,t}\}$: expression (2.2)

3. critère d'arrêt: $e = \sum_{i=1}^c \|\mathbf{v}_{i,t} - \mathbf{v}_{i,t-1}\|$

Tant que $e > \varepsilon$

L'algorithme HCM / FCM converge depuis toute initialisation vers un minimum local [3]. Le cas $m = 1$ correspond à une coalescence stricte, et $m = \infty$ correspond à $u_{ik} \rightarrow (1/c) \forall i, k$ (équid'appartenance d'une observation à toutes les classes). En pratique, il n'existe pas des règles permettant de fixer m , mais le choix heuristique $m = 2$ donne généralement des résultats satisfaisants [6].

Il est aussi clair que \mathbf{U} sera plus significative lorsque l'ensemble d'apprentissage est de taille suffisamment importante pour être considéré comme représentatif de l'ensemble des échantillons possibles.

2.5. Classification

L'objectif est de concevoir une fonction de décision permettant de classifier/étiqueter une nouvelle observation \mathbf{x}_{n+1} . Voici une discussion sur les critères disponibles.

Critère 1: Règle des k -plus proches voisins: kppv (k -NN rule)

- 1) Charger $\mathbf{X} = \{\mathbf{x}_i\}$, ensemble d'apprentissage; et \mathbf{U} la matrice de partition correspondante (coalescence floue),
- 2) Choisir une valeur pour k : nombre des plus proches voisins,
- 3) Choisir $d: \mathfrak{R}^q \times \mathfrak{R}^q \rightarrow \mathfrak{R}^+$ une métrique (mesure de distance) quelconque sur \mathfrak{R}^q ,
- 4) Pour une nouvelle observation $\mathbf{z} \notin \mathbf{X}$:
 - Calculer/ordonner les distances $d(\mathbf{z}, \mathbf{x}_i)$ sous la forme: $\{d_1 \leq d_2 \leq \dots \leq d_k \leq d_{k+1} \leq \dots \leq d_n\}$.
 - Trouver les colonnes dans \mathbf{U} correspondant aux indices des k plus proches voisins: $\{1, 2, \dots, k\}$.
 - Calculer le vecteur $\mathbf{u}(*|\mathbf{z}) = (u(1|\mathbf{z}), u(2|\mathbf{z}), \dots, u(c|\mathbf{z}))^T$, avec

$$u(i|\mathbf{z}) = \sum_{j=1}^k \frac{u_{ij}}{k}; \text{ pour } i = 1, 2, \dots, c \quad (2.3)$$

- Décision: $D_{\text{knn}}(\mathbf{z}) = \mathbf{e}_i \Leftrightarrow u(i|\mathbf{z}) = \max_j \{u(j|\mathbf{z})\}$. (\mathbf{e}_i élément d'une base orthogonale qui désigne la classe i).

Critère 2: Règle du plus proche prototype (NP rule)

$$\mathbf{z} \in i \Leftrightarrow D_{\text{NP}, \mathbf{v}}(\mathbf{z}) = \mathbf{e}_i \Leftrightarrow \|\mathbf{z} - \mathbf{v}_i\| \leq \|\mathbf{z} - \mathbf{v}_j\| \text{ pour } j = 1, \dots, c. \quad (2.4)$$

Critère 3:

$$\mathbf{z} \in i \Leftrightarrow \mu_i(\mathbf{z}) = \max \{\mu_j(\mathbf{z})\}_{j=1, \dots, c}. \quad (2.5)$$

$\mu_j(\mathbf{z})$ est estimé par un approximateur non linéaire dont l'optimisation fait appel aux techniques d'apprentissage des Réseaux de Neurones Artificiels (chapitre 3)

Nous présentons dans le chapitre suivant un outil permettant la conception des fonctions d'appartenance, permettant donc au classifieur de s'adapter aux observations hors apprentissage.