الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DR. YAHIA FARES DE MEDEA

FACULTE DES SCIENCES ET DE LA TECHNOLOGIE

Département du Génie Électrique
et de l'Informatique



Mémoire de Fin d'Études

Présenté pour l'obtention du Diplôme
de Master en Informatique

Option : Ingénierie des Systèmes et Technologies du Web

THEME

Analyse des données issues de puces à ADN par la méthode de classification SVM

Proposé et dirigé par :

M. Brahim SAHMADI

Préparé et présenté par :

- Melle. YAHIAOUI Djalila
- Melle. AZIZI Souad

Remerciements

Au terme de ce travail nous tenons à remercier tout d'abord ALLAH le tout puissant de nous avoir donné le courage et d'avoir facilité la réalisation de ce travail.

Nous voulons exprimer par ces quelques lignes de remerciements nos gratitudes envers tous ceux en qui, participer à ce travail par leur présence, leur soutien, leur disponibilité et leurs conseils.

Nous exprimons notre grande reconnaissance et remerciement à notre promoteur M. SAHMADI Brahim pour son aide, ces conseils et de nous avoir orienté et mis à notre disposition tous les moyens nécessaires durant notre travail.

En fin, nous ne pouvons achever ce projet sans exprimer notre Gratitudes à tous les enseignants de l'Université Dr. Yahia FARES -MEDEA-, pour leur dévouement et leur assistance tout au long de cette année.

Dédicace

Je dédie ce travail à la lumière
de ma vie : Mes chers parents, que nulle dédicace
ne peut exprimer mes sincères sentiments, pour leur patience
Illimitée, leur encouragement contenu, leur aide, en
Témoignage de mon profond amour et respect
Pour ses grands sacrifices.
A mon Marie pour leur soutien, encouragement
et de la patience et de l'aide continue
pendant cette année
A mon cher frère et mes chères sœurs,
Pour leur grand amour
Et leur soutien qu'ils trouvent ici
L'expression de ma haute gratitude.
A mes chers amies et A mon binôme
qui sans leur encouragement ce travail
N'aura jamais vu le jour.
Et à toutes les familles « YAHIAOUI et BELHNNICHE
A tous ceux qui ont participé de près ou de loin à la réalisation
de ce projet je dédie le fruit de ce mémoire.

Djalila

Dédicace

Au Nom de Dieu clément et miséricordieux,

Louange à Dieu pour sa Grâce et sa miséricorde,

Louange à Dieu pour la science qui nous a offerte,

Du fond du cœur;

Je dédie ce modeste travail à :

A Mes très chers parents,

Mes frères et à Ma seul sœur zola , Ma nouvelle famille ,Mon marie,et tous les

Membres de la famille,

Ma binôme yahiaoui Djalila et sa famille,

A tous mes amis,

Sans oublier mon promoteur et mes professeurs.

A toutes personnes

Ayant contribué de loin ou de près

À accomplir ce travail.

souad

Résumé :

Le problème de la sélection de variables en classification se pose généralement lorsque le nombre de variables est très élevé. Néanmoins si l'on doit traiter des données décrites par un grand nombre de variables, les méthodes classiques d'analyse, d'apprentissage ou de fouille de données peuvent se révéler inefficaces ou peuvent conduire à des résultats peu précis. Dans ce mémoire, nous proposons une méthode innovante basée sur une méta-heuristique de recuit simulé combinée avec la méthode de classification SVM, pour réduire la taille initiale des données et pour sélectionner des ensembles de variables pertinents pour le but d'une classification supervisée des données issues de puces d'ADN.

Mots-clés : sélection de gènes, classification, optimisation combinatoire, recuit simulé, puces à ADN.

Abstract

The problem of selection of classification variables usually arises when the number of variables is very high. However if one has to deal with data described by many variables, the classical methods of analysis, learning or data mining may be ineffective or may lead to inaccurate results.

In this work, we propose an innovative method based on simulated annealing meta-heuristic combined with SVM classification method, to reduce the initial size of the data and to select sets of relevant variables for the purpose of a supervised classification of DNA Microarray data.

Key Words : gene selection, classification, combinatorial optimization, simulated annealing, DNA Microarray.

ملخص :

مشكلة اختيار المتغيرات في عمليات تحليل المعطيات تنشأ عندما يكون عدد المتغيرات مرتفع جدا. واذا تم التعامل مع البيانات بالطرق التقليدية للتحليل قد تكون النتائج غير دقيقة.

في هذه المذكرة, نقترح طريقة جديدة تعتمد على خوارزمية التبريد المقلد و طريقة التصنيف SVM للحد من الحجم الأولي للبيانات واختيار مجموعات من المتغيرات ذات الصلة بتحليل البيانات الجينية.

كلمات مفتاحية : اختيار الجينات, تصنيف, الاندماج الامثل, خوارزمية التبريد المقلد, رقائق الحمض النووي.

Sommaire

Introduction générale 1

Chapitre I: Technologie des puces à ADN

I.1 Introduction : 4
I.2 Généralités : 4
ADN 4
Gènes 4
ARN messager 6
Traduction 6
Protéines 6
Mutation 6
Gène de maladie : 7
I.3 Puces d'ADN : 8
I.3.1 Historique : 8
I.3.2 Principaux types de puces à ADN : 9
1.3.3 Phases d'analyse des puces à ADN : 11
Fabrication de la puce : 11
Préparation de la cible : 12
Hybridation de la cible et de la sonde : 12
Lecture brute des données : 12
Présentation des données de puces à ADN : 12
I.4 Application variées pour les puces à ADN : 13
I.5 Conclusion : 14

Chapitre II: Classification et sélection d'attributs

II.1 Introduction : 17
II.2 Fouille de données : 17
II.2.1 Définitions 18
1.2.2 Processus d'extraction de connaissances 19
II.2.3 Tâches de la fouille de données : 20
II.3 Classification : 21
II.3.1 Buts et modalités de la classification : 21

II.3.2 Classification supervisée : 22
II.3.3 Méthodes de classification supervisée : 23
II.3.3.1SVM : (Machines à Vecteurs de Support) : 23
II.3.3.2 Arbres de décision 23
II.3.3.3 Réseaux de neurones : 24
II.3.3.4 K plus proches voisins : 24
II.3.4 Classification non supervisée : 25
II.3.5 Méthodes de classification non supervisée : 26
II.4 Sélection des attributs : 26
II.4.1 Notions de pertinence, non pertinence et redondance : 27
II.4.1.1 Pertinence d'attributs : 27
II.4.1.2 Redondance d'attributs : 28
II.4.1.3 Processus global de la sélection de variables 30
II.4.1.4 Les approches pour la sélection d'attributs 31
II.5 Conclusion 33

Chapitre III: Recuit simulé et SVM pour la sélection des gènes et la classification

III.1Introduction : 36
III.2 Recuit simulé : 36
III.2.1 Organigramme de recuit simulé : 37
III.2.2 Les grandes lignes de l'algorithme de recuit simulé : 38
III.3 SVM (Machines à Vecteurs de Support) : 38
III.3.1 Principe de fonctionnement général de SVM : 39
III.3.1.1 Notions de base : Hyperplan, marge et support vecteur 39
III.3.1.2 Pourquoi maximiser la marge ? 40
III.3.1.3 Linéarité et non-linéarité 41
III.3.1.3.1 Cas non linéaire 42
III.3.1.3.2 Illustration de transformation de cas non linéaire : le cas XOR 42
III.3.2 SVM multi classe ……..43
III.4 Méthode d'enveloppe proposée : 44
III.4.1 Organigramme : 45
III.4.2 Architecture : 46

III.5 Conclusion : 47

Chapitre IV: Conception du système de prédiction

IV.1 Introduction : 49
IV.2 Fonctionnalités du système : 49
IV.2.1 Diagramme des cas d'utilisation : 49
IV.2.2 Description des cas d'utilisation : 50
IV.2.3 Diagrammes d'activités : 50
IV.3 Description des outils utilisés : 52
IV.3.1 LIBSVM : 52
IV.3.2 Builder C++ XE4 : 57
IV.4 Structure technique du système : 57
IV.4.1 Couches du système : 58
IV.4.2 Diagramme de classes : 59
IV.5 Conclusion : 59

Chapitre V: Implémentation et résultats

V.1 Introduction : 61
V.2 Aperçu de l'application développée : 61
V.3 Expériences et résultats : 63
V.3.1 Présentation des bases de données utilisées : 63
V.3.2 Prétraitement des données : 64
V.3.2 Sélection des paramètres : 64
V.3.3 Résultats : 65
5.4 Conclusion : 68
Conclusion générale 69
Bibliographie 70

Liste de figures

Chapitre I

Figure I.1 : Modèle de la double hélice d'ADN…….5
Figure I.2 : Du gène à la protéine : transcription des gènes en ARNm (transcriptome) et traduction des ARNm en protéines (protéome)……..7
Figure I.3. Macroarrays après hybridation de l'échantillon (taille : 12 cm x 8 cm)……...9
Figure I.4. Microarray après hybridation de l'échantillon (taille : 5,4 cm x 0,9 cm)……10
Figure I.5 : GeneChipAffymetrix (taille : 2,5 cm x 7,5 cm x 1 cm)…….....11
Figure I.6 : Les étapes d'une analyse par puces à ADN……...11

Chapitre II

Figure II.1 : Différentes étapes du processus ECD……..19
Figure II.2: Catégorisation des variables……..29
Figure II.3: Processus de sélection de variables……...30
Figure II.4: L'approche filtre……32
Figure II.5 : Principe de l'approche wrapper et embedded……..…33

Chapitre III

Figure III.1 : L'organigramme de l'algorithme du recuit simulé……...37
Figure III.2 : Les grandes lignes de l'algorithme de la technique de recuit simulé……..38
Figure III.3 : un hyperplan……39
Figure III.4 : L'hyperplan séparateur optimal……..40
Figure III.5 : hyperplan optimal et hyperplan avec faible marge……...….41
Figure III.6 : Linéarité et non-linéarité…….41
Figure III.7 : Cas non linéaire vers cas linéaire……42
Figure III.8 : Le cas XOR (dimension 2)……..…43
Figure III.9 : Le cas XOR (dimension 3)…….…43
Figure III.10 : Exemples SVM multiclasses…….44
Figure III.11: Organigramme de la méthode proposée…….45
Figure III.12 Architecteur de développement……...46

Chapitre IV

Figure IV.1 : Diagramme de cas d'utilisation……...49
Figure IV.2 : Diagramme d'activité : chargement des données d'apprentissage …….50
Figure IV.3 : Diagramme d'activité : paramétrage de l'algorithme de recuit simulé……...51
Figure IV.4 : Diagramme d'activité : choix des paramètres du SVM……..51
Figure IV.5 : Diagramme d'activité : calcule et affichage de résultats……52.
Figure IV.3 : Découpage de l'application de prédiction en 3 couches logiques……...58
Figure IV.4 : Diagramme de classes…….59

Chapitre V

Figure V.1. Interface de l'application…… ..62
Figure V.2. Exemple de fichier modèle……63
Figure V.3 : Taux de classification pour les données Breast_cancer……...……66
Figure V.4 : Taux de classification pour les données Heart…….67
Figure V.5 : Taux de classification pour les données Diabète…….68

Liste des tableaux

Chapitre I

Table I.1 : Matrice d'expression des gènes……..……13

Chapitre V

Table V.1 : Taux de classification pour les données Breast_cancer……....65

Table 5.2 : Taux de classification pour les données Heart……..…66

Table 5.3 : Taux de classification pour les données Diabète …….…....67

Introduction générale

La bio-informatique est une discipline relativement récente, le terme ayant été créé dans les années 80. Cette notion englobe l'ensemble des applications de l'informatique aux sciences de la vie, domaine très vaste qui recouvre tous les axes de recherche, allant des applications en robotique aux techniques les plus avancées en intelligence artificielle. Pour la plupart des membres de la communauté scientifique, cette notion semble dans la pratique s'adapter plus particulièrement aux outils informatiques qui permettent de stocker, d'analyser et de visualiser les informations contenues dans les séquences des gènes et des protéines des êtres vivants. L'histoire de la bio-informatique est donc étroitement liée à celle de la biologie moléculaire et l'étude des molécules du vivant.

Ce travail s'intéresse aux données issues des puces à ADN. Cette technologie permet de mesurer simultanément les niveaux d'expression de gènes au sein d'échantillons de tissus dans des conditions expérimentales données. Les premiers jeux de données du domaine oncologique ont été publiés à la fin des années 90. Le travail de Golub par exemple a montré que les seules données issues des puces à ADN permettaient de discriminer deux formes de leucémie. De plus, parmi les quelques 7000 gènes testés sur les puces pour cette expérience, un petit nombre de gènes (environ 50) apparaît comme très important pour la reconnaissance des deux formes de la maladie. Depuis le début des années 2000, un grand nombre de travaux se sont intéressés au problème de la classification des données issues des puces à ADN avec l'espoir de proposer des outils de diagnostic des différents cancers et aussi de compréhension des mécanismes de ces pathologies.

Comme les données analysées présentent plusieurs milliers d'attributs, il est nécessaire de proposer des méthodes innovantes pour la réduction de dimension ou la sélection de gènes.

Le problème de la sélection d'attributs consiste à choisir les attributs les moins redondants et les plus pertinents pour accomplir la tâche de classification et ce mémoire propose une méthode pour réaliser au mieux cette sélection.

Ce mémoire est structuré en cinq chapitres. Les deux premiers ont pour objectif d'exposer le contexte et la problématique de la sélection de variables ainsi que les méthodes

utilisés dans cet essor. Les trois derniers chapitres sont dédiés à nos contributions dans ce sujet.

Chapitre1 : dans ce chapitre nous introduisons la technologie des puces à ADN et nous montrons l'importance et la nécessité de la sélection de variables dans ce domaine.

Chapitre 2 : dans ce chapitre nous introduisons et nous exposons les techniques de fouille de données et de la classification, la présentation des concepts de base d'un problème de sélection de variables et les notions nécessaires à la construction d'un algorithme de sélection de variables.

Chapitre 3 : dans ce chapitre nous présentons notre méthode hybride de sélection de variables basée sur le recuit simulé et la méthode de classification supervisée SVM

Chapitre 4 : nous donnons dans ce chapitre la conception de la méthode proposée en utilisant la notation UML

Chapitre 5 : consiste à présenter l'interface graphique et les résultats expérimentaux.

Nous terminons ce mémoire par une conclusion général et quelques perspectives qui peuvent donner suite à ce travail.

CHAPITRE I

Technologie des puces à ADN

I.1 Introduction :

La recherche en génétique progresse très vite, le séquençage du génome est maintenant fini pour de nombreuses espèces comme les levures ou les souris... et même le génome humain n'aurait plus de secrets. Cependant, la connaissance des séquences codantes de tous les gènes d'un organisme nécessite le développement de nouvelles biotechnologies pour étudier tous les gènes. Une étude des gènes un par un n'est pas concevable à cause de leur grand nombre : environ 6000 gènes pour les levures. La technique des biopuces est l'un des outils les plus puissants qui a émergé à la suite du séquençage du génome.

Pour comprendre comment marchent les biopuces à ADN, il est nécessaire de donner quelques explications biologiques sur la structure de l'ADN et sur les différents acteurs de l'activité de la cellule**.**

I.2 Généralités :

Tout organisme est constitué d'une ou plusieurs cellules. Chacune de ces cellules contient de l'ADN qui peut être localisé dans le noyau chez les eucaryotes, ou dans le cytoplasme chez les procaryotes.

ADN (Acide Désoxyribonucléique) (figure I.1) est une longue molécule faite de deux chaînes de poly-nucléotides enroulées en spirale autour d'un axe imaginaire pour former une double hélice. Les bordures externes de la double hélice sont formées par la succession de pentoses-phosphates et les bases azotées (adénine (A), thymine (T), guanine (G) et cytosine (C)) se font face à l'intérieur de la double hélice. L'adénine d'un brin forme toujours deux ponts hydrogène avec une thymine qui lui fait face sur l'autre brin, et pareillement, la guanine d'un brin forme toujours trois ponts hydrogène avec la cytosine de l'autre brin. Donc, la séquence d'un brin détermine la séquence de l'autre.

Gènes :Un gène est une portion d'ADN qui contient toute la recette d'assemblage d'une protéine. Ce sont les protéines qui font l'essentiel du travail dans l'organisme et les gènes permettent leur reproduction de génération en génération. Concrètement, un gène est une

longue séquence de A, C, G et T dans l'ADN. Une partie de cette longue séquence contient les « plans de construction » de la protéine en langage génétique (Chaque combinaison de trois lettres génétiques correspond à un acide aminé particulier, c'est à dire à une des «briques» dont sont faites les protéines.)**[1].**

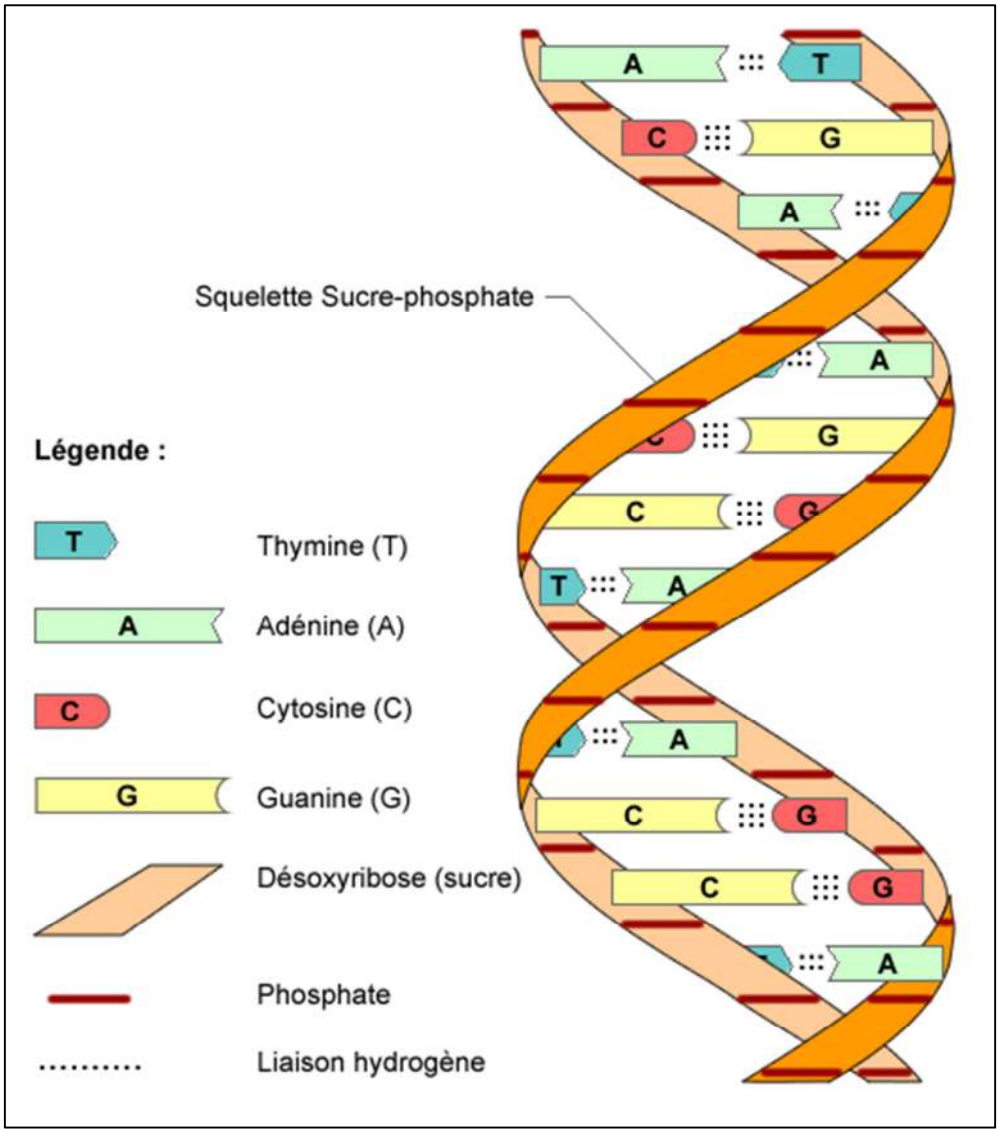


Figure I.1 : Modèle de la double hélice d'ADN

Le reste de la séquence d'un gène indique à la cellule dans quelle circonstance elle doit exécuter les instructions de ce gène. Par exemple, l'insuline est fabriquée dans le pancréas. Le gène de l'insuline, contient donc une séquence spécifique de A, C, G et T qui attire la « tête de lecture » du pancréas pour enclencher le décodage du gène. Même si le même gène de l'insuline est aussi présent dans les cellules des autres organes, ceux-ci n'ont pas la bonne tête de lecture et ne peuvent donc pas l'utiliser.

Selon les individus, il peut exister plusieurs versions d'un même gène, appelées allèles. Par exemple, nous avons tous des gènes pour la couleur de nos yeux mais certains allèles donnent des yeux bleus et d'autres, des yeux bruns.

ARN messager (Acide Ribonucléique) : est comme l'ADN une chaine composée de nucléotides, ou l'Uracile (U) remplace la Thymine, c'est une « photocopie » de l'ADN produite durant la phase de transcription. En effet pour chaque gène, l'ADN qui le compose est transcrit en une molécule complémentaire qui est l'ARN. Le noyau de la cellule est comme une bibliothèque ou sont stockées les informations et comme dans toutes la bibliothèque, les originaux ne peuvent sortir donc l'ADN est transcrit en un nombre variable d'exemplaires d'ARN messager sans jamais quitter le noyau de la cellule, c'est l'ARN qui transporte l'information génétique à travers le reste de la cellule.

Traduction : La traduction = synthèse de la protéine à partir de l'ARN messager.

L'enchaînement des nucléotides de l'ARN messager est décodé dans les ribosomes par triplet : 3 nucléotides = 1 codon.

Protéines **:** en fonction de ces besoins, la cellule va utiliser tout ou une partie des gènes pour réaliser la synthèse des protéines nécessaires aux fonctions cellulaires. Durant la phase de traduction l'ensemble des ARN messagers quittent le noyau et sont distribués dans la cellule pour la protection de protéine correspondante (figure I.2).

Après la traduction, l'enchaînement des 20 acides aminés de la protéine correspond exactement à celui des codons de l'ARN messager.

Mutation **:** Une mutation est simplement une modification accidentelle dans l'ADN. Cela peut être une lettre remplacée par une autre (Ex: un A par un G) ou même un long morceau d'ADN qui se voit sectionné, déplacé et collé à un autre. Toutes sortes de phénomènes peuvent provoquer des mutations. Les radiations ionisantes n'en sont qu'un exemple.

Beaucoup de mutations n'ont aucun effet sur l'organisme. Même si une mutation tuait une cellule, celle-ci serait rapidement remplacée par une cellule neuve. Parfois une accumulation de mutations peut amener une cellule à se multiplier de façon anarchique : c'est un cancer.

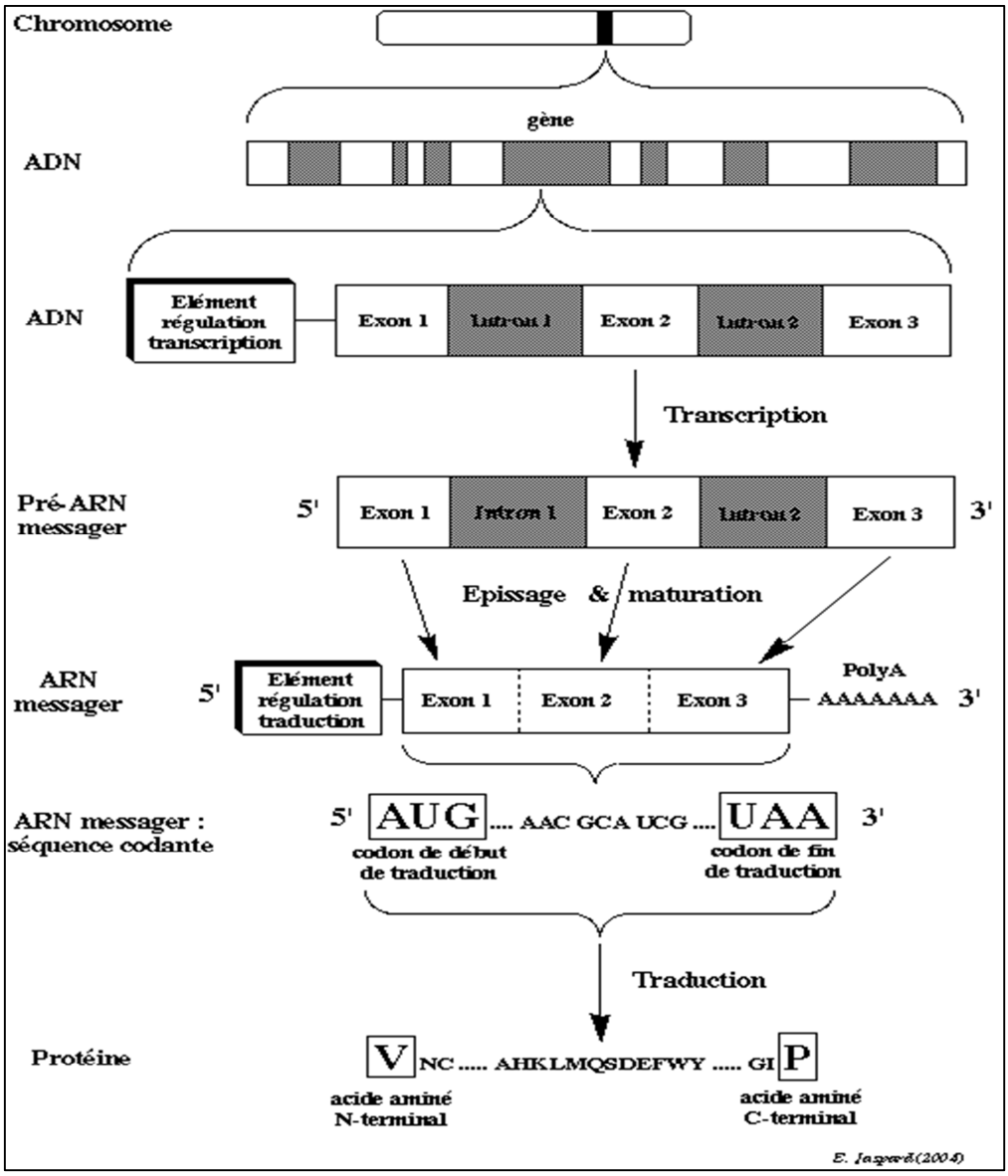


Figure I.2 : Du gène à la protéine : transcription des gènes en ARNm (transcriptome) et traduction des ARNm en protéines (protéome) [2].

Gène de maladie :Le gène d'une maladie ne s'attrape pas dans l'air comme une infection. Cela veut simplement dire qu'une personne a reçu à la naissance un gène défectueux (ou un assortiment nuisible de plusieurs gènes) qui le plus vulnérable à une maladie particulière.

I.3 Puces d'ADN :

Les puces d'ADN ou les biopuces ont été développées au début des années 1990. Depuis leur apparition, les puces à ADN sont devenues des outils majeurs pour la recherche en biologie fondamentale et clinique. Elles permettent de mesurer et visualiser très rapidement les différences d'expression entre plusieurs milliers de gènes dans une cellule ou un tissu donné, à partir d'un support solide de taille réduite (quelques cm^2) sur lequel des milliers de fragment d'ADN sont immobilisés selon une disposition ordonnée à l'aide d'une micropipette robotisée.

Grâce à cette technique, chacun des fragments d'ADN est représenté par un point ou « spot »[1] sur le support, ils servent de sondes[2] pour fixer de façon très spécifique les fragments des gènes complémentaires « cibles »[3], présents dans les échantillons biologique à tester : leur mise en contact permet de reconstituer la double hélice d'ADN. Ce phénomène repose sur le principe d'hybridation entre des séquences complémentaires d'ADN. L'emplacement de séquence est repéré par la position (x_i, y_i) correspond au gène i.

Les puces à ADN sont des lames de verre activé sur lesquelles sont déposés de nombreuses copies d'une séquence d'ADN spécifique d'un gène donné.

I.3.1 Historique :

Les premières puces à ADN sont apparues en 1993, mais leur concept date de 1987. Suite logique aux anciennes méthodes de northernblotting et d'expression différentielle, la technologie des puces à ADN est basée sur le principe d'hybridation développé par Southern (1974). Ce principe stipule que deux fragments d'acides nucléiques complémentaires peuvent s'associer et se dissocier de façon réversible sous l'action de la chaleur et de la concentration saline du milieu. Il s'agit simplement d'une miniaturisation du système classique de reverse dot blot qui a vu le jour grâce à une technologie pluridisciplinaire intégrant l'électronique (technique de dépôt), la chimie (préparation des lames et greffes des sondes oligonucléotidiques ou synthèse in situ), l'analyse d'image (acquisition des données) et

[1] Un ***spot*** correspond à de **nombreuses sondes**, c'est-à-dire à de nombreuses copies d'une séquence d'ADN spécifique d'un gène donné.
[2] Les sondes **:**Les acides nucléiques fixés sur les puces à ADN sont appelés **sondes.**
[3] Les cibles : Les acides nucléiques qui sont hybridés avec ces sondes sont appelés cibles.

l'informatique (interprétation des données). Depuis leur apparition, les puces à ADN suscitent un intérêt inversement proportionnel à leur taille [3].

I.3.2 Principaux types de puces à ADN :

Il existe aujourd'hui trois grands types de puces à ADN : les macroarrays, les microarrays et les puces à oligonucléotides (cette nomenclature pouvant varier).

1. Pour la construction des macroarrays ou filtres à haute densité, des gouttelettes d'ADNc sont déposées sur une membrane de nylon. Il peut y avoir jusqu'à 2.400 gouttelettes déposées sur une membrane, chacune étant constituée d'un grand nombre de copies d'une séquence d'ADNc. Une fois déposées sur la membrane, les gouttelettes ont une taille comprise entre 0,5 et 1 mm. L'échantillon, à tester par ce type de puce à ADN, doit être de l'ADNc. Une seule condition expérimentale peut êtretestée par puce. La figure I.3 montre un macroarray après hybridation de l'échantillon. Plusun spot (unité d'hybridation) est grand, plus l'échantillon a pu s'hybrider à cet endroit, et donc, plus le gène fixé à cet endroit de la membrane était exprimé dans l'échantillon.

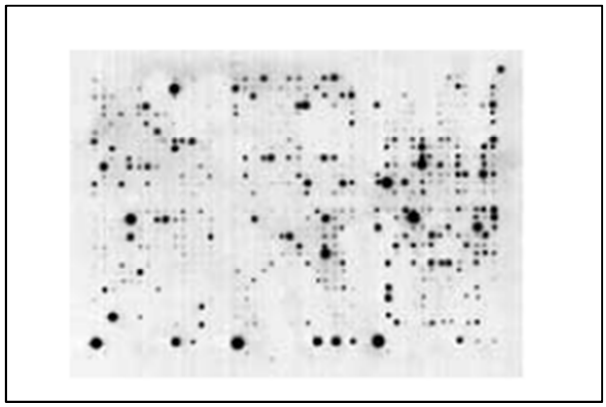


Figure I.3 :Macroarrays après hybridation de l'échantillon (taille : 12 cm x 8 cm)[18]

2. Les microarrays ont été développés en 1997, lors de leur fabrication, des gouttelettes contenant un grand nombre de copies d'une séquence de 30 à 70 nucléotides d'ADNc, chacune, sont déposées sur une lame de verre traitée par un revêtement chimique qui permet de fixer l'ADN. Jusqu'à 10.000 gouttelettes peuvent être déposées par cm^2de support, le diamètre de chaque spot mesurant environ 100 µm. Ce type de puce à ADN demande de l'ADNc comme échantillon, celui-ci doit être marqué à la cyanine 3 pour

une condition et à la cyanine 5 pour l'autre. L'un des avantages de cette technique est donc de pouvoir tester deux conditions sur une même puce. Par contre, elle ne donne qu'une mesure relative du taux d'expression des gènes entre les deux conditions. Sur la figure I.4, on peut voir un microarray après hybridation de l'échantillon. Les spots rouges montrent que les gènes fixés à ces endroits de la puce sont plus exprimés dans la condition A que dans la condition B, les spots verts montrent que les gènes fixés à ces endroits de la puce sont plus exprimés dans la condition B que dans la condition A, enfin, les spots jaunes montrent que les gènes fixés à ces endroits de la puce sont exprimés de manière équivalente dans les conditions A et B.

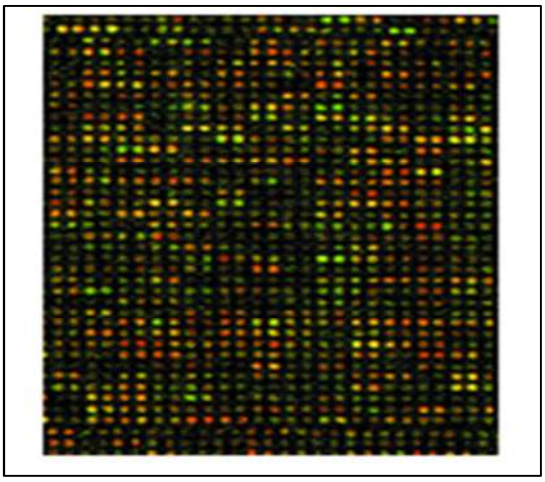


Figure I.4 : Microarray après hybridation de l'échantillon (taille : 5,4 cm x 0,9 cm)[18]

3. Les puces à oligonucléotides sont des lames de verre à revêtement chimique où des séquences d'ADNc de 20 à25 nucléotides sont synthétisées par photolithographie ou par impression de type « jet d'encre» .Ce sont les puces à ADN présentant la plus haute densité de spots, puisqu'il peut y en avoir plus de 250.000 par cm^2 de support, chacun ayant une taille d'environ 20 µm. Les puces à oligonucléotides nécessitent de l'ARN marqué à la biotine-streptavidine comme échantillon. Une seule condition peut être testée par puce, mais en plus de pouvoir représenter tout un génome grâce à la haute densité de spots, les puces à oligonucléotides ont l'avantage de donner des valeurs absolues d'expression des gènes. En effet, dans ce cas, il n'y a pas de compétition entre deux qui déterminent la couleur du spot. La couleur est toujours la même, seule son intensité diffère en fonction de l'expression des gènes. La figure I.5 montre une puce à oligonucléotides Affymetrix ou GeneChip[1].



Figure I.5 :GeneChip Affymetrix (taille : 2,5 cm x 7,5 cm x 1 cm) [18]

1.3.3 Phases d'analyse des puces à ADN :

Les différentes phases d'une analyse par puces à ADN sont indiquées dans la (figure I.6) :

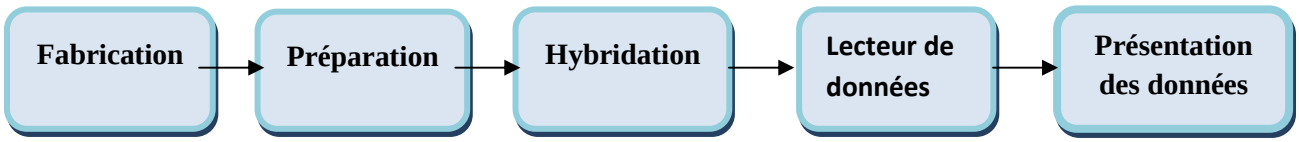


Figure I.6: Les étapes d'une analyse par puces à ADN.

Fabrication de la puce : Les fragments d'ADN dont on souhaite déterminer le niveau d'expression sont amplifiés par la technique de PCR (Polymérase Chain Réaction) puis déposés sur une lamepréalablement recouverte de poly lysine (cette substance assure la fixation de l'ADN via des interactions électrostatiques). Ces fragments d'ADN fixés à un support constituent des sondes. Ces puces sur lame de verre sont appelées microarrays. Les lieux de dépôts de l'ADN sur le support solide sont des spots**.**

L'ADN est dénaturé pour qu'il se trouve sous la forme simple brin sur la puce, ce qui lui permettra ultérieurement de s'associer au brin complémentaire contenu dans la cible.

Préparation de la cible :Des ARN sont extraits de différents types de cellules (notés arbitrairement ici cellules 1et cellules 2) pour lesquelles on veut comparer le niveau d'expression du gène étudié (c'est-à-dire celui qu'on a fixé sur la puce). Les ARN messagers permettent d'obtenir des ADNc par transcription inverse.

Les ADNc sont ensuite marqués de manière différente : par exemple, l'ADN des cellules 1 est marqué avec un fluorochrome vert, tandis que l'ADN des cellules 2 est marqué avec un fluorochrome rouge.

Hybridation de la cible et de la sonde :Les **ADN marqués** en vert et ceux marqués en rouges sont **mélangés** (cet ensemble constitue la **cible**) puis **déposés sur la sonde**. La puce est alors **incubée** une nuit à 60 degrés : à cette température, un brin d'ADN s'apparie avec son complémentaire ce qui forme de l'ADN double brin. **Les ADN fluorescents vont ainsi s'hybrider sur les gènes qui avaient été déposés sur la sonde.**

Lecture brute des données :Chaque **spot** est excité par un laser et**on enregistre la fluorescence** émise d'une part dans le rouge, d'autre part dans le vert. On obtient alors deux images dont le niveau de gris représente l'intensité de la fluorescence lue. En remplaçant les niveaux de gris par des niveaux de vert pour la première image et des niveaux de rouge pour la seconde, on obtient en les superposant une **image en fausses couleurs composée de spots allant du vert**(cas où seulement de l'ADN des cellules 1 s'est fixé à la sonde) **au rouge** (cas où seulement de l'ADN des cellules 2 s'est fixé à la sonde) **en passant par le jaune** (cas où l'ADN des deux types de cellules s'est fixé en quantité égale).

Présentation des données de puces à ADN :Les données recueillies pour l'étude d'un problème donné sont regroupées sous forme de matrice avec une ligne par couple (gène, sonde) et une colonne par échantillon. Chaque valeur m_{ij} de la matrice est la mesure du niveau d'expression du ième gène dans le jème échantillon, oùi= 1,…,Met j= 1,…,N.

$Gène_{id}$	$Echantillon_1$	$Echantillon_2$	⋯	$Echantillon_N$
$Gène_1$	m_{11}	m_{12}	⋯	m_{1N}
$Gène_2$	m_{21}	m_{22}	⋯	m_{2N}
$Gène_3$	m_{31}	m_{32}	⋯	m_{3N}
⋮	⋮	⋮	⋯	⋮
$Gène_M$	m_{M1}	m_{M2}	⋯	m_{MN}

Table I.1: Matrice d'expression des gènes.

Les différentes phases d'une analyse par puces ADN peuvent être source d'imprécision ou d'erreurs dans les mesures obtenues. De plus, le coût d'une puce à ADN et le coût d'une analyse étant très élevé et l'on ne dispose à l'heure actuelle que de quelques dizaines d'expériences pour l'étude d'un problème donné (une pathologie par exemple). Pourtant chaque expérience a permis de relever le niveau d'expression pour plusieurs milliers de gènes. Les matrices de données qui sont actuellement disponibles ont donc les caractéristiques suivantes :

- Grande dimensionnalité due au nombre élevé de descripteurs (gènes) ;
- Nombre limité d'échantillons.

De ce fait, la sélection des gènes est une tâche importante et nécessaire pour une meilleure exploitation des données des puces à ADN [4].

I.4 Application variées pour les puces à ADN :

Il existe aujourd'hui de nombreuses applications dans lesquelles les puces à ADN sont impliquées, les plus importantes sont brièvement évoquées ici, mais il en existe bien d'autres.

- **Mesure de la transcription des gènes :** est la première et principale application des puces à ADN. En hybridant un échantillon à une puce à ADN, on obtient une mesure de la quantité des ARNm présents dans l'échantillon. En effectuant cette mesure dans différentes conditions ou sur différents types cellulaires, on peut identifier les gènes corrégulés dans certaines réponses cellulaires spécifiques, prédire la fonction de gènes non caractérisés, ou encore mettre en évidence des réseaux de régulation de voies biochimiques complexes.

- **Criblage de mutations :** appelé aussi SNP pour single Nucléotide polymorphisme, repose sur la conception de puces d'ADN capables d'analyser chaque base d'une séquence connue, pour cela les sondes sont organisées en ensembles au sein desquels une séquence est strictement homologue à celles du type sauvage alors que les autres sont caractérisées par une substitution de base toujours localisées au milieu de la séquence, par exemple, pour étudier une substitution de base, les sondes sont organisées en tétrades dans lesquelles une des quatre sondes possède une position centrale la base homologue à celle de la séquence sauvage, alors que trois autre sonde contiennent les trois autres bases possibles.

- **Séquençage par hybridation :** Une autre application des puces d'ADN a été développée pour le Séquençage par hybridation, son utilisation permet d'accélérer considérablement le procédé. Le gène étudie est en effet considéré comme un ensemble de plusieurs séquences chevauchantes dont la détermination simultanée puis l'assemblage permettent de reconstruire la séquence (Hacia ,1999). La méthode est également utilisée pour la vérification de génomes déjà séquencé par la technique de Sanger.

- **Application industrielles :** Les puces trouvent également de nombreuses applications dans l'industrie des puces conçues pour des cibles d'ARN ribosomiaux sont ainsi utilisées pour identifier les souches pathogènes dans les hôpitaux. De mêmes elles sont utilisées dans le secteur agro-alimentaire pour le contrôle qualité (contrôle des micro-organismes utilisées dans certaines fabrications, détection des séquences provenant d'organismes génétiquement modifiés) et pour l'environnement (détection des agents infectieux dans l'alimentation, l'air ou l'eau).

  Les puces à ADN sont aussi un outil de diagnostic. Notamment, elles permettent une nouvelle classification plus fine des types de cancers. De plus, grâce aux puces à ADN, on peut identifier de nouveaux gènes cibles de substances thérapeutiques, ainsi que la réponse cellulaire à un traitement [5].

I.5 Conclusion :

Nous avons présenté dans ce chapitre les notions qui s'articulent autour de la technologie des puces à ADN, comme nous avons pu le constater elles offrent de nombreux espoirs dans le domaine médical .En oncologie, le développement du profil génique de l'ensemble des

tumeurs définies permettre une amélioration du diagnostic, et la fabrication de nouveaux médicaments. Désormais couramment utilisées en clinique. Cependant quelques difficultés voient le jour concernent son utilisation. D'une part d'un point de vue économique, leur cout est bien trop élevé. D'autre part, au niveau pratique, il faut automatiser leur utilisation, et développer une analyse bien plus précise.

CHAPITRE II

Classification et sélection d'attributs

II.1 Introduction :

La fouille de données (ou datamining) consiste à rechercher et à extraire de l'information, utile et inconnue, à partir de gros volumes de données stockées dans des bases ou des entrepôts de données. Le développement récent de la fouille de données (depuis le début des années 1990) est lié à plusieurs facteurs : une puissance de calcul importante est disponible sur les ordinateurs ; le volume des bases de données augmente énormément ; l'accès aux réseaux de taille mondiale, ces réseaux ayant un débit sans cesse croissant, qui rendent le calcul distribué et la distribution d'information sur échelle mondiale variable. La fouille de données a aujourd'hui une grande importance économique du fait qu'elle permet d'optimiser la gestion des ressources humaines et matérielles.

La classification est la tâche la plus importante de la fouille de données qui consiste à examiner des caractéristiques d'un objet afin de l'affecter à une classe d'un ensemble donné. Par ailleurs, il y a lieu de rappeler que l'utilisateur qui veut couvrir tous les aspects existants d'un problème particulier et obtenir une connaissance compréhensible doit considérer un grand nombre de variables. Or, parmi ces variables certaines sont inutiles. En effet, il est souvent difficile voire impossible de discerner les variables pertinentes des variables non pertinentes ce qui pousse l'utilisateur à s'emparer de toutes les variables disponibles. De plus, les sources de données peuvent être multiples et la fusion des données issues de chacune de ces sources conduit à la création d'un ensemble contenant des variables inutiles et redondantes.

La solution que l'on peut apporter à cette difficulté et la sélection d'un sous ensemble de variables. La sélection de variables et un processus permettant l'élimination des variables inutiles et/ou redondantes et l'élimination du bruit pouvant être généré par certaines variables. Le processus de fouille de données est accéléré et la précision prédictive des algorithmes de classification peut être améliorée.

II.2 Fouille de données :

Depuis quelques années, une masse grandissante de données sont générées de toute part par les entreprises, que ce soit des données bancaires, telles que les opérations de carte de crédit, ou bien des données industrielles, telles que des mesures de capteurs sur une chaîne de

production, ou toutes autres sortes de données possibles et imaginables. Ce flot continu et croissant d'informations peut être maintenant stocké et préparé à l'étude, grâce aux nouvelles techniques d'entrepôt de données. Les techniques usuelles d'analyse de données, développées pour des tableaux de tailles raisonnables ont largement été mises à mal lors de l'étude de tant de données. En effet, alors que le principal objectif de la statistique est de prouver une hypothèse avancée par un expert du domaine, et donc de confirmer une connaissance déjà connue ou bien présumée, le but de la fouille de données est maintenant de découvrir, au sens propre du terme, des nouvelles connaissances. Et ceci sans faire appel à des hypothèses préconçues. Ce nouveau concept de fouille de données, bien qu'il paraît révolutionnaire pour certains, est en fait une autre vision et une autre utilisation de méthodes existantes, et combinées.

Ainsi, au vue de l'émergence de ces deux champs d'application (fouille et entrepôt de données), une idée nouvelle s'est faite. Pourquoi ne pas associer toutes ces techniques afin de créer des méthodes puissantes de recherche de connaissances, intégrant toutes les étapes, du recueil des données à l'évaluation de la connaissance acquise. C'est ainsi qu'est né le terme d'Extraction des Connaissances à partir des Données (ECD), ou en anglais Knowledge Discovery in Database (KDD).

II.2.1 Définitions

L'extraction des connaissances à partir de données consiste à parcourir d'immenses volumes de données contenus dans une base, à la recherche de connaissances. C'est une discipline qui se situe à l'intersection de différents domaines tels que l'informatique, l'intelligence artificielle, l'analyse de données, les statistiques, la théorie des probabilités, l'optimisation, la reconnaissance de formes, les bases de données et l'interaction Homme-Machine,… Il est ici important de différencier les trois termes suivants :

- Donnée : valeur d'une variable pour un objet (comme le montant d'un retrait d'argent par exemple) ;
- Information : résultat d'analyse sur les données (comme la répartition géographique de tous les retraits d'argent par exemple) ;
- Connaissance : information utile pour l'entreprise (comme la découverte du mauvais emplacement de certains distributeurs).

Ainsi à l'aide de l'ECD, nous pouvons à partir de données sur lesquelles nous ne faisons aucune hypothèse, obtenir des informations pertinentes,et de celles-ci, tirer des connaissances.

1.2.2 Processus d'extraction de connaissances

Le terme processus signifie que l'ECD se décompose en plusieurs opérations (voir Figure II.1).

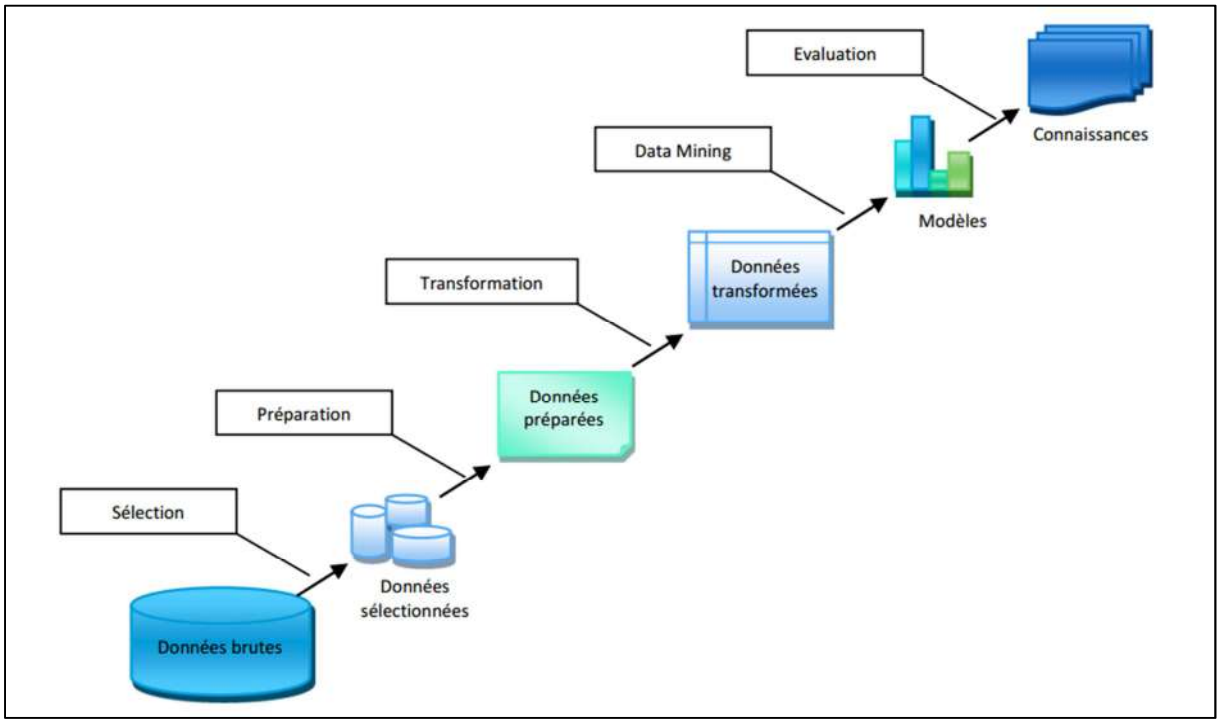


Figure II.1 : Différentes étapes du processus ECD

Ces opérations peuvent être regroupées en cinq phases majeures :

- Compréhension du domaine étudié : Lors de cette phase, une analyse du problème et des contraintes qui lui sont attachées doit permettre la collecte de données brutes. Ces données se composent d'individus ou objets et des variables qui leurs sont associées et qui doivent permettre de décrire au mieux le problème traité. L'utilisateur ne sait pas encore si les données qu'il a réunies seront toutes adaptées à son problème ni si ces données seront suffisantes. Nous sommes en présence des données initiales.
- Prétraitement : Lors de cette phase, un prétraitement est effectué à la fois sur les individus et sur les variables. Cette phase de prétraitement consiste à nettoyer les données, les mettre en forme, traiter les données manquantes, échantillonner les individus, sélectionner et construire des variables. On obtient ainsi un ensemble de données cibles. Cette phase a une place importante au sein du processus d'ECD car

c’est elle qui va déterminer la qualité des modèles construits lors de la phase de fouille de données. Elle peut prendre jusqu’à 60% du temps dédié au processus d’ECD.

- Fouille de données : Cette phase intègre le choix de la méthode d’apprentissage qui va être employée et son paramétrage. Ces choix doivent tenir compte des contraintes liées au domaine étudié ainsi que des connaissances que les experts du domaine peuvent nous fournir. L’algorithme sélectionné est alors appliqué aux données cibles dans le but de rechercher les structures sous-jacentes des données et de créer des modèles explicatifs ou prédictifs. Certes la fouille de données n’est qu’une étape du processus de l’ECD, mais elle est sans conteste le cœur et le moteur de tous ce processus.
- Post-traitement : Cette phase consiste en l’évaluation et la validation des modèles construits lors de la phase précédente. Ce n’est qu’après cette phase que les données et l’information que l’on en a tirée deviennent des connaissances.
- Interprétation et exploitation des résultats : L’interprétation des résultats qui sont sous forme de modèles ou de règles permet d’obtenir des connaissances. Ce sont ces connaissances qui seront fournies à l’utilisateur.

La finalité de l’ECD est de pouvoir traiter des données brutes et volumineuses, et à partir de ces données, d’établir des connaissances directement utilisables par un expert ou un non expert du domaine étudié.

II.2.3 Tâches de la fouille de données :

Les principales tâches de fouille de données sont :

- La classification (classification supervisé) : consiste à examiner des caractéristiques d’un objet afin de l’affecter à une classe d’un ensemble prédéfini. Les classes sont discrètes ;
- L’estimation : permet d’obtenir une variable continue en combinant les données en entrée. L’estimation est souvent utilisée pour effectuer une tâche de classification en utilisant un barème ;
- La prédiction : ressemble à la classification et à l’estimation, mais les enregistrements sont classés selon un certain comportement futur prédit ou à une valeur future estimée. s’appuie sur le passé et le présent mais son résultat se situe dans un futur généralement précisé.

- Le regroupement par similitudes : consiste à déterminer les objets qui vont naturellement ensemble ;
- La segmentation (classification automatique) : consiste segmenter une population hétérogène en sous-populations homogènes. Contrairement à la classification, les sous populations ne sont pas préétablis
- La description : il s'agit de décrire les données d'une base complexe. Cette tâche engendre souvent une exploitation supplémentaire en vue de fournir des explications [6].

II.3 Classification :

L'apprentissage et la classification automatique (Machine Learning) ont été conçu pour la résolution automatique des problèmes complexes par la prise de décisions sur la base d'observations de ces problèmes.

En effet, le progrès scientifique a permis d'augmenter sensiblement les capacités de mesures et a ainsi augmenté la taille des bases de données. Il est à présent difficile pour un opérateur humain de traiter ces données dans un temps raisonnable. Ce qui rendre la tâche de prise de décision plus difficile et complexe pour un opérateur humain.

II.3.1 Buts et modalités de la classification :

La classification repose sur des objets à classer. Les objets sont localisés dans un espace de variables (ont dit aussi attributs, caractéristiques ou critères). Il s'agit de les localiser dans un espace de classes. Ce problème n'a de sens que si on pose l'existence d'une correspondance entre ces deux espaces. Résoudre un problème de classification, c'est trouver une application de l'ensemble des objets à classer, décrits par les variables descriptives choisies, dans l'ensemble des classes. L'algorithme ou la procédure qui réalise cette application est appelé classifieur.

Nous appellerons :

- classificateur : une règle établie (estimée) de classification, c'est-à-dire une fonction sur l'espace des caractéristiques vers l'espace des classes ;
- classification : la construction d'un classificateur ;
- classement : la mise en œuvre d'un classificateur existant.

Généralement, l'inférence statistique traditionnelle peut couvrir plusieurs problématiques :

- exploratoire : déceler des relations hypothétiques ;
- prédictive : valider la performance globale d'un système de relations ;
- explicative : valider des composantes détaillées d'un système de relations· comprendre leurs contributions à ce système.

On retrouve des distinctions voisines en classification :

- On appelle classification automatique, ou non supervisée, un ensemble de problématiques où l'espace des classes n'est pas spécifié à l'avance. Il s'agit d'identifier, voire de construire, un système de classes sur la base d'observations dans l'espace des caractéristiques.
- On appelle classification supervisée un contexte où un ensemble de classes (et une structure sur cet ensemble) est spécifié à l'avance.

II.3.2 Classification supervisée :

Dans le cadres de la classification supervisée, des données d'apprentissage labellisées sont disponibles afin d'apprendre les règles de décisions qui permettront de distinguer les classes de données. Ces observations, dites d'apprentissage, supervisent la construction du classifieur. La classification supervisée se base sur deux aspects principaux : descriptif et décisionnel. L'aspect descriptif essai de trouver une représentation qui permet l'interprétation des groupes grâce aux variables explicatives.

Quant à l'aspect décisionnel, on cherche à définir la meilleure affectation d'un nouvel individu dont on ne connaît que les valeurs des variables explicatives qui le décrivent. Cet aspect est particulièrement intéressant dans des domaines ou la notion de diagnostic est essentielle.

Apprentissage supervisé : Dans ce cas, chacune des données utilisées dans l'apprentissage appartient à une des K classes. Les données $x_1,\ldots,x_n$, décrites par p variables explicatives et appartenant aux différentes K classes $y_1,\ldots,y_n$ seront ainsi représentées par un ensemble de couples (observation, labels)sous la forme suivante : $(x_1,y_1),\ldots,(x_n,y_n)$,

Les règles de décision seront alors générées à partir des informations contenues dans les labels des classes et les attributs [7].

II.3.3 Méthodes de classification supervisée :

II.3.3.1SVM : (Machines à Vecteurs de Support) :

Les machines à vecteurs de support (Support Vector Machine, SVM) appelés aussi séparateurs à vaste marge sont des techniques d'apprentissage supervisées destinées à résoudre des problèmes de classification. Les machines à vecteurs supports exploitent les concepts relatifs à la théorie de l'apprentissage statistique et à la théorie des bornes de Vapnik et Chervonenkis. La justification intuitive de cette méthode d'apprentissage est la suivante : si l'échantillon d'apprentissage est linéairement séparable, il semble naturel de séparer parfaitement les éléments des deux classes de telle sorte qu'ils soient le plus loin possibles de la frontière choisie. En plus, de nombreux développements ont été réalisés pour proposer des variantes traitant le cas non-linéaire. Le succès de cette méthode est justifié par les solides bases théoriques qui la soutiennent. Elles permettent d'aborder des problèmes très divers dont la classification. SVM est une méthode particulièrement bien adaptée pour traiter des données de très haute dimension [8].

II.3.3.2 Arbres de décision

Les arbres de décision sont une des méthodes les plus connues en classification. Le principe des arbres de décision est de réaliser la classification d'un exemple par une suite de tests sur les attributs qui le décrivent. Concrètement, dans la représentation graphique d'un arbre,

1. Un nœud interne correspond à un test sur la valeur d'un attribut.
2. Une branche part d'un nœud et correspond à une ou plusieurs valeurs de ce test.
3. Une feuille est un nœud d'où ne part aucune branche et correspond à une classe.

Une règle de décision (de la forme si . . . alors . . .) est créée pour chaque chemin partant de la racine de l'arbre et parcourant les tests (en faisant des conjonctions) jusqu'à la feuille qui est l'étiquette de la classe. Les arbres de décision sont particulièrement appréciés car ils permettent une compréhension aisée, mais lors d'une tâche de classification relativement complexe l'utilisateur n'est plus capable d'explorer efficacement les résultats obtenus sous forme textuelle.

Les principaux algorithmes de construction d'un arbre de décision sont : C4.5,CHAID, et CART. Dans ces algorithmes, à chaque étape de création d'un nœud, un critère de séparation

entre les classes est utilisé pour décider que l'attribut est le plus pertinent pour la classification [5].

II.3.3.3 Réseaux de neurones :

Le principe consiste à apprendre à classifier correctement des données à partir d'un jeu d'exemples déjà classifiés, c'est-à-dire l'apprentissage par l'expérience. Un réseau de neurones est constitué d'un graphe pondéré orienté dont les nœuds symbolisent les neurones. Ces neurones possèdent une fonction d'activation qui permet d'influencer les autres neurones du réseau, les fonctions, les plus souvent utilisées, sont la fonction signe ou la fonction sigmoïde.

Les connexions entre les neurones, que l'on nomme liens synaptiques, propagent l'activité des neurones avec une pondération caractéristique de la connexion. On appelle poids synaptique la pondération des liens synaptiques.

Les neurones peuvent être organisés de différentes manières, c'est ce qui définit l'architecture et le modèle du réseau. L'architecture la plus courante est celle dite des perceptrons multicouches.

Points positifs : excellentes performances, minimise la fréquence d'erreur. Point négatif : difficulté de mise en œuvre [9].

II.3.3.4 K plus proches voisins :

L'algorithme des k plus proches voisins (noté k-PPV) est une méthode basée sur la notion de proximité (voisinage) entre exemples et sur l'idée de raisonner à partir de cas similaires pour prendre une décision. Autrement dit des entrées xi semblables devraient avoir des valeurs yi semblables. Le principe est le suivant : on note x un nouvel exemple décrit par un vecteur de p attributs. On trouve alors, parmi l'ensemble d'exemples d'apprentissage, les k plus proches voisins de x et on associe à x la classe majoritaire parmi ses k voisins lui ressemblant le plus dans la base d'apprentissage. Cette méthode dépend donc des trois éléments suivants :

1. Le nombre de voisins retenus.
2. La mesure de distance entre exemple.
3. La combinaison des classes.

Le résultat dépend du réglage de ces paramètres. Pour le premier critère, on utilise généralement un nombre de voisins compris entre 1 et 7. Pour le deuxième paramètre, la méthode nécessite une métrique pour mesurer la proximité entre l'exemple à classifier x et chacun des exemples de l'ensemble d'apprentissage. Lorsque les attributs sont numériques la distance euclidienne est généralement utilisée. Le troisième paramètre indique de quelle manière on combine les valeurs associées aux voisins pour obtenir la valeur associée à x. la combinaison des classes des k plus proches voisins en une classe C. Pour la classification, la classe retenue pour x est la classe majoritaire chez ses voisins. Les points positifs de cette méthode sont qu'elle ne pose aucune hypothèse sur la forme des classes à apprendre. La méthode est simple puisqu'il n'y a pas besoin d'apprentissage d'un modèle de classification et son pouvoir prédictif est souvent bon.

Mais la performance de cette méthode diminue lorsque la dimension augmente, puisque pour chaque nouvelle classification, il est nécessaire de calculer toutes les distances de x à chacun des exemples d'apprentissage. De plus, la performance dépend fortement de k, le nombre de voisins choisi et il est nécessaire d'avoir un grand nombre d'observations pour obtenir une bonne précision des résultats [5].

II.3.4 Classification non supervisée :

La principale différence entre la classification non supervisée ou le clustering, et la classification supervisée, est que le jeu de données ne comprend aucune information sur la distribution des données d'apprentissage. Autrement dit, le jeu d'apprentissage ne contient que les observations $x_1,\ldots,x_n$ et les labels $y_1, \ldots,y_n$, alors que le nombre des classe est manquant[5].

Apprentissage non supervisé :

Dans le cadre de l'apprentissage non supervisé, les classes des données ne sont pas connues à priori : les labels $y_1, \ldots,y_n$ ne sont pas donnés. Donc le jeu de données ne peut être représenté qu'avec les valeurs $x_1,\ldots,x_n$ décrites eux-mêmes par les p variables explicatives. L'apprenant ou bien le classificateur ne peut se baser que sur la notion de similarité entre attributs pour pouvoir retrouver les règles de décision [7].

II.3.5 Méthodes de classification non supervisée :

La classification non supervisée ou "clustering" est l'une des techniques fondamentales de l'extraction de données structurées ou non structurées. Plusieurs méthodes ont été proposées:

- Classification hiérarchique : arbre de classes
- Classification hiérarchique ascendante : Agglomérations successives.
- Classification hiérarchique descendante : Divisions successives.
- Classification à plat : algorithme des k-moyennes : Partition.

II.4 Sélection des attributs :

La sélection d'attributs est un sujet de recherche très actif depuis une dizaine d'années dans les domaines de l'apprentissage artificiel, de la fouille de données, du traitement d'images, et de l'analyse de données en bioinformatique .Elle consiste à choisir parmi un ensemble d'attributs de grande taille, un sous-ensemble d'attributs intéressants pour le problème étudié. Cette problématique peut concerner différentes tâches d'apprentissage ou de fouille de données, mais nous parlerons dans ce travail de la sélection d'attributs réalisée pour la classification supervisée.
Dans ce contexte, les principales motivations de la sélection d'attributs sont les suivantes :

1. Utiliser un sous-ensemble plus petit permet d'améliorer la classification si l'on élimine les attributs qui sont source de bruit. Cela permet aussi une meilleure compréhension des phénomènes étudiés.
2. Des petits sous-ensembles d'attributs permettent une meilleure généralisation des données en évitant le sur-apprentissage.
3. Une fois que les meilleurs attributs sont identifiés, les temps d'apprentissage et d'exécution sont réduits et en conséquence l'apprentissage est moins coûteux.

En présence de centaines voire de milliers d'attributs il y a beaucoup de chances pour que des attributs soient corrélés et expriment des informations similaires, on dira alors qu'ils sont redondants. D'un autre côté, les attributs qui fournissent le plus d'information pour la classification seront dits pertinents. L'objectif de la sélection est donc de trouver un sous-ensemble optimal d'attributs qui ait les propriétés suivantes : il doit être composé d'attributs pertinents et il doit chercher à éviter les attributs redondants. De plus cet ensemble doit

permettre de satisfaire au mieux l'objectif fixé c'est-à-dire la précision de l'apprentissage, la rapidité de l'apprentissage ou bien encore l'explicabilité du classifieur proposé.

Les notions de pertinence et redondance jouent un rôle fondamental dans la sélection d'attributs et nous allons donc les détailler dans la section suivante [10].

II.4.1 Notions de pertinence, non pertinence et redondance :

Nous présentons d'abord une formalisation de la pertinence des attributs qui permet de distinguer les différentes contributions d'un attribut à la définition d'une fonction de classification. De même, nous présentons une définition de la redondance qui permet de proposer un mécanisme d'élimination des attributs redondants.

II.4.1.1 Pertinence d'attributs :

Les spécialistes définissent la pertinence dans le cas d'attributs et de fonctions booléennes, et en supposant que les données sont non bruitées [13]. Une définition plus large proposée dans [14] définit les attributs pertinents comme ceux dont les valeurs varient systématiquement avec les valeurs de classe. Autrement dit, un attribut Fi est pertinent si connaître sa valeur change les probabilités sur les valeurs de la classe C. Mais cette définition peut être précisée pour distinguer les attributs fortement pertinents et les attributs faiblement pertinents grâce aux définitions suivantes. Soit F un ensemble complet d'attributs, F_i un attribut, et $S_i = F - \{F_i\}$. On suppose que l'on travaille avec un espace probabilisé où la probabilité est notée P. P(C|S) est la probabilité de la classe C connaissant les attributs de l'ensemble S.

Définition 2.1 Un attribut Fi est fortement pertinent ssi :

$$P(C|Fi, Si) \neq P(C|Si)$$

Définition 2.2 Un attribut Fi est faiblement pertinent ssi :

$$P(C|F_i, S_i) = P(C|S_i) \text{ et } \exists S'_i \subset S_i \text{ tel que } P(C|F_i, S'_i) \neq P(C|S'_i)$$

Définition 2.3 Un attribut Fi est non pertinent ssi :

$$P(C|F_i, S_i) = P(C|S_i) \text{ et } \forall S'_i \subseteq P(C|F_i, S'_i) = P(C|S'_i)$$

D’après ces définitions, les variable fortement pertinentes sont donc indispensables et devraient figurer dans tout sous-ensemble optimal sélectionné, car leurs absences peuvent conduire à un défaut de reconnaissance de la fonction cible (la classe).
La faible pertinence suggère que la variable n’est pas toujours importante, mais il peut devenir nécessaire pour un sous-ensemble optimal dans certaines conditions.
La non-pertinence d’une variable se définit simplement par rapport à (2.1) et (2.2) et indique qu’une variable n’est pas du tout nécessaire dans un sous-ensemble optimal de variables.

II.4.1.2 Redondance d’attributs :

La notion de la redondance d’attributs se comprend intuitivement et elle est généralement exprimée en termes de corrélation entre attributs. On peut dire que deux attributs sont redondants (entre eux) si leurs valeurs sont complètement corrélées. Cette définition ne se généralise pas directement pour un sous-ensemble d’attributs. On trouve dans [15], une définition formelle de la redondance qui permet de concevoir une approche pour identifier et éliminer les attributs redondants. Cette formalisation repose sur la notion de couverture de Markov (Markov blanket) d’un attribut qui permet d’identifier les attributs non pertinents et redondants.

Définition 2.4 : Soit F l’ensemble total d’attributs et C la classe. Soit F_i un attribut, et M_i un sous-ensemble d’attributs qui ne contient pas F_i ,c’est-à-dire : $M_i \subseteq F$ et $F_i / \in M_i$ M_i est une couverture de Markov pour Fi ssi :

$$P (F - M_i - \{ F_i \} , C \mid F_i, M_i) = P (F - M_i - \{ F_i \} , C \mid M_i)$$

La définition de couverture de Markov impose que M_i subsume non seulement l’information que F_i apporte sur C mais aussi l’information qu’il apporte sur tous les autres attributs. Dans [15], il est montré qu’un sous-ensemble d’attributs optimal peut être obtenu par une procédure d’élimination descendante, appelée filtrage par couverture de Markov (Markov blank et Filtering) et définie comme suit :

Filtrage par couverture de Markov :
Soit G l’ensemble d’attributs courant (G = F au départ). A chaque étape de la procédure, s’il existe une couverture de Markov pour l’attribut F_i dans l’ensemble G courant, Fi est enlevé de G.
On peut montrer que ce processus garantit qu’un attribut enlevé dans une étape précédente peut trouver une couverture de Markov dans une étape postérieure.

Selon les définitions précédentes de la pertinence d'attributs, on peut également montrer que les attributs fortement pertinents ne peuvent trouver aucune couverture de Markov. Par contre, les attributs non pertinents doivent être enlevés de toute façon, et il n'est donc pas nécessaire de s'y intéresser dans la définition des attributs redondants. Cela conduit à la définition suivante de la redondance [16].

Définition 2.5 : Soit G l'ensemble d'attributs courant, un attribut F_i est redondant et par conséquent peut être enlevé de G ssi il est faiblement pertinent et qu'il possède une couverture de Markov dans G.

Dans le schéma de la figureII.2, un ensemble initial de variables peut être partitionné en quatre catégories : variables non pertinentes (partie I), variables redondantes (partie II) (qui sont faiblement pertinentes comme on l'a vu), variables faiblement pertinentes et non-redondantes (partie III), et variables fortement pertinentes (partie IV).

Un sous-ensemble de variables optimal contient essentiellement toutes les variables des parties III et IV. Il est important de préciser que pour un ensemble initial donné, le processus de filtrage par couverture de Markov peut conduire à différents découpages donnant les parties II et III (qui sont disjointes) [16].

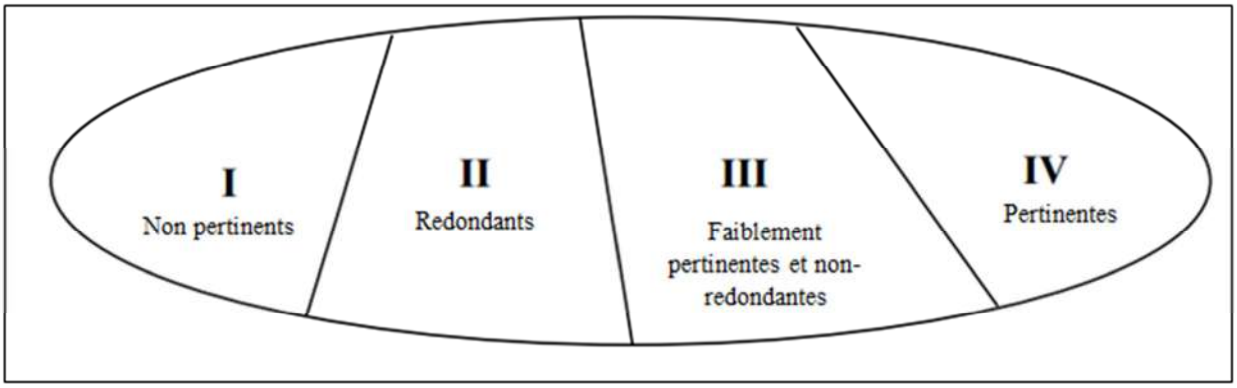


Figure II.2 : Catégorisation des variables

Les formalisations précédentes sont intéressantes pour mieux cerner les notions de redondance et de pertinence. Néanmoins les définitions probabilistiques que nous avons vues ne permettent pas de proposer un processus de sélection d'attributs applicable sur des données de grande dimension. Dans la suite de ce chapitre nous allons donc présenter différents points de vue permettant de comprendre le processus de la sélection d'attributs, d'un point de vue plus opérationnel [12].

II.4.1.3 Processus global de la sélection de variables :

Les différentes méthodes proposées dans la littérature pour la sélection de variables peuvent être décrites par un schéma général (voir Figure II.3) dans lequel on trouve les éléments clés suivants :

- Une procédure de génération de sous-ensembles candidats qui détermine l'exploration de l'espace de recherche ;
- Une fonction d'évaluation donnant la qualité des sous-ensembles candidats ;
- Une condition d'arrêt ;
- Un processus de validation pour vérifier si l'objectif souhaité est atteint [6].

Schéma général de la sélection d'attributs :

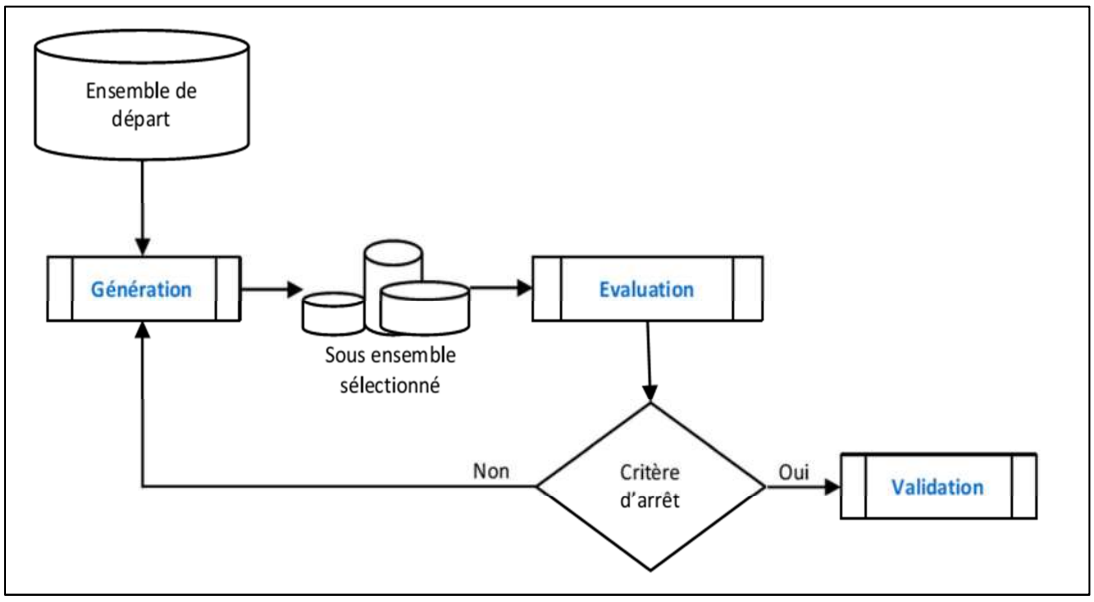


Figure II.3 : Processus de sélection de variables.

La sélection de variables comporte trois éléments essentiels : une mesure de pertinence, une procédure de recherche et un critère d'arrêt [6].

Mesure de pertinence : La mesure de pertinence associée aux méthodes de sélection de variables sont souvent basées sur des heuristiques calculant l'importance individuelle de chaque variable dans le modèle obtenu, ces heuristiques sont de différentes natures :

statistique, probabiliste, information mutuelle ou celle qui mesure l'indépendance ou la vraisemblance entre les variables.

Procédure de recherche : Trouver une solution optimale suppose une recherche exhaustive parmi les (2^{n-1}) sous-ensembles de variables possibles, et bien que des méthodes de recherche efficaces comme le Branch and Bound ont été proposé, elles s'appuient sur une propriété de mono tonicité de la mesure de pertinence qui est en pratique difficile à assurer. Une recherche exhaustive est dès lors inapplicable même pour un nombre de variables de l'ordre de quelques dizaines [17].

Critère d'arrêt : L'initialisation et le critère d'arrêt définissent les bornes de la recherche. Dans le cas de méthodes basées sur un critère d'évaluation individuelle la condition d'arrêt peut être un nombre fixé d'attributs à retenir. Dans le cas des méthodes d'évaluation de sous-ensembles, le critère d'arrêt peut être un temps de calcul fixé, un nombre d'itérations fixé, l'absence de gain de performance par rapport aux solutions déjà trouvées ou encore le fait que les sous-ensembles candidats deviennent trop homogènes (dans le cas d'algorithmes à base de populations) [5].

II.4.1.4 Les approches pour la sélection d'attributs **:**

Le schéma général présenté dans la section précédente se retrouve dans les 3 grands types d'approches que nous décrivons ci-dessous. Ces approches se distinguent par la manière dont la sélection d'attributs interagit (ou non) avec le mécanisme de classification.

- **Méthodes de Filtres :**

L'approche filtre sélectionne un sous ensemble de variables en-prétraitement des données d'un modèle (l'étape de l'analyse des données), Le processus de sélection est indépendant du processus de classification (Figure II.4). Un de ces avantages est d'être complètement indépendant du modèle de données que nous cherchons à construire. Elle propose un sous ensemble de variables satisfaisant pour expliquer la structure des données qui se cachent et que le sous ensemble est indépendant de l'algorithme d'apprentissage choisi. Ce contexte est aussi adaptatif dans la sélection de variables non supervisées. De plus les procédures filtres

sont généralement moins couteuses en temps de calcul puisqu'elles évitent lesexécutions répétitives des algorithmes d'apprentissage sur différents sous ensemble de variables.

En revanche, leur inconvénient majeur est qu'elles ignorent l'impact des sous-ensembles choisis sur les performances de l'algorithme d'apprentissage.



Figure II.4: L'approche filtre.

- **Méthodes Enveloppes (**Wrapper**):**

Leurs principe est de générer des sous-ensembles candidats et de les évaluer grâce à un algorithme de classification. Cette évaluation est faite par un calcul d'un score, par exemple un score d'un ensemble sera un compromis entre le nombre de variables éliminées et le taux de réussite de la classification sur un fichier de test. L'appel de l'algorithme de classification est fait plusieurs fois à chaque évaluation (c'est-à-dire à chaque sélection d'une variable, nous calculons le taux de classification pour juger la pertinence d'une caractéristique) car un mécanisme de validation croisée est fréquemment utilisé. Le principe de wrappers est de générer un sous ensemble bien adaptés à l'algorithme de classification. Les taux de reconnaissance sont élevés car la sélection prend en compte le biais intrinsèque de l'algorithme de classification. Un autre avantage est sa simplicité conceptuelle ; nous n'avons pas besoin de comprendre comment l'induction est affectée par la sélection des variables, il suffit de générer et de tester.

Cependant, trois raisons font que les wrappers ne constituent pas une solution parfaite .D'abord, ils n'apportent pas vraiment de justification théorique à la sélection et ils ne nous permettent pas de comprendre les relations de dépendances conditionnelles qu'il peut y avoir entre les variables.

D'autre part la procédure de sélection est spécifique à un algorithme de classification particulier et les sous-ensembles trouvés ne sont pas forcément valides si nous changeons de méthode d'induction. Finalement, c'est l'inconvénient principale de la méthode, les calculs devient de plus en plus très longs, voir irréalisables lorsque le nombre de variables est très grand.

- **Méthodes Embedded**

LesméthodesEmbeddedintègrentdirectementlasélectiondansleprocessusdel'apprentissage (FigureII.5), les arbres de décision sont l'illustration la plus emblématique. Mais, en réalité, nous classons dans ce groupe toutes techniques qui évaluent l'importance d'une variable en cohérence avec le critère utilisé pour évaluer la pertinence globale du modèle [5].

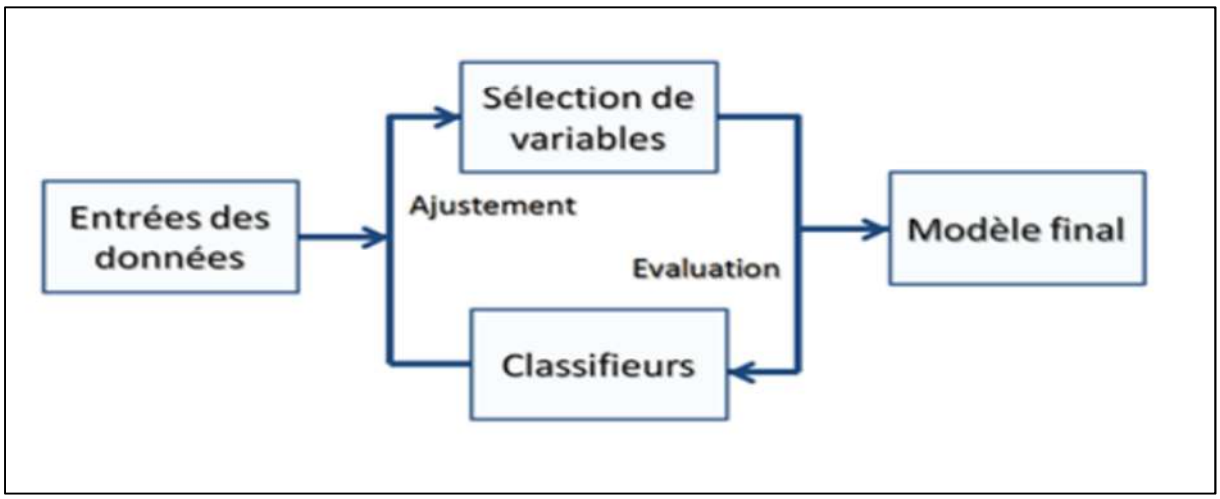


Figure II.5 : Principe de l'approche wrapper et embedded.

II.5 Conclusion

Dans ce chapitre nous avons présenté la fouille de données qui constitue le cœur du processus d'extraction de connaissances à partir des données. Ensuite nous avons présenté la classification et ces méthodes. Nous avons vu aussi dans ce chapitre que la sélection d'attributs est une étape de prétraitement qui joue un rôle important dans la fouille de données. Elle permet de représenter un sous ensemble de données à partir d'un ensemble volumineux de données et d'éliminer les données redondantes, non pertinentes ou bruitées. Il y a plusieurs avantages de la sélection de sous ensemble d'attributs : Elle facilite la visualisation des données et fournit une meilleure compréhension. Elle réduit la complexité de données d'apprentissage qui va conduire à la réduction du temps de l'algorithme d'apprentissage. Un autre facteur important est la réduction de la dimension du problème, l'amélioration de la performance de la prédiction et la compréhension du modèle d'apprentissage. Ceci est réalisé en supprimant les attributs non pertinents à partir de l'ensemble total des attributs en préservant les avantages mentionnés ci-dessus. Appliquée à la tâche de la classification supervisée, la sélection d'attributs améliore la précision et la

compréhension du classifieur. La recherche d'un sous ensemble d'attributs est un problème d'optimisation NP-difficile qui peut être résolu par les méta-heuristiques.

CHAPITRE III

Recuit simulé et SVM pour la sélection des gènes et la classification

III.1Introduction :

Les problèmes NP-complets d'optimisation combinatoire sont caractérisés par une complexité exponentielle ou factorielle, par conséquent ; il est impossible d'énumérer toutes les solutions possibles car cela dépasse la capacité de calcul de n'importe quel ordinateur. Il est donc très difficile de trouver la solution optimale.

Pour pallier à ces problèmes, les chercheurs ont introduit des méthodes approchées appelées heuristiques, elles présentent l'avantage d'un temps de calcul réduit mais ne donnent aucune information sur la qualité de la solution trouvée, de plus elles ne sont en général applicables qu'a un seul type de problèmes.

Depuis une vingtaine d'années, les heuristiques les plus populaires, et également les plus efficaces, sont des techniques générales, appelées méta-heuristiques, qu'il s'agit d'adapter à chaque problème particulier. Parmi ces techniques générales, nous nous pencherons dans ce chapitre sur une approche ayant clairement démontré leur utilité dans de nombreux domaines. Cette approche est appelée Recuit Simulé.

III.2 Recuit simulé :

Le recuit simulé est une méthode de recherche de solution optimale, dont le mécanisme de recherche est calqué sur le principe du recuit thermodynamique. Le refroidissement progressif d'un système de particules est simulé en faisant une analogie entre l'énergie du système et la fonction objectif du problème (P) d'une part, et entre les états du système et les solutions admissibles de (P) d'autre part (Voir l'organigramme de la FigureIII.1). Pour atteindre des états avec une énergie aussi faible que possible, on porte initialement le système à très haute température puis on le refroidit petit à petit. Tout nouvel état est obtenu en faisant subir un déplacement infinitésimal aléatoire à un atome quelconque. Soit ΔE la différence d'énergie occasionnée par une telle perturbation. Le nouvel état est accepté si l'énergie du système diminue (ΔE < 0). Dans le cas contraire (ΔE = 0), il est accepté avec une certaine probabilité : $prob(\Delta E, t) = \exp(\frac{-\Delta E}{K_B t})$ où t est la température du système et k_B une constante physique connue sous le nom de constante de Boltzmann. A chaque étape, l'acceptation d'un nouvel état dont l'énergie n'est pas inférieure à celle de l'état courant est décidée en générant de

manière aléatoire un nombre q∈ [0,1[. Si q est inférieur ou égal à prob(ΔE,t), alors le nouvel état est accepté. Autrement l'état courant est maintenu. L'utilisation répétée d'une telle règle fait évoluer le système vers un état d'équilibre thermique.

En règle générale, la température est diminuée par paliers à chaque fois qu'un certain nombre d'itérations est effectué. L'algorithme est interrompu lorsqu'aucune solution voisine n'a été acceptée pendant un cycle complet d'itérations à température constante [11].

III.2.1 Organigramme de recuit simulé :

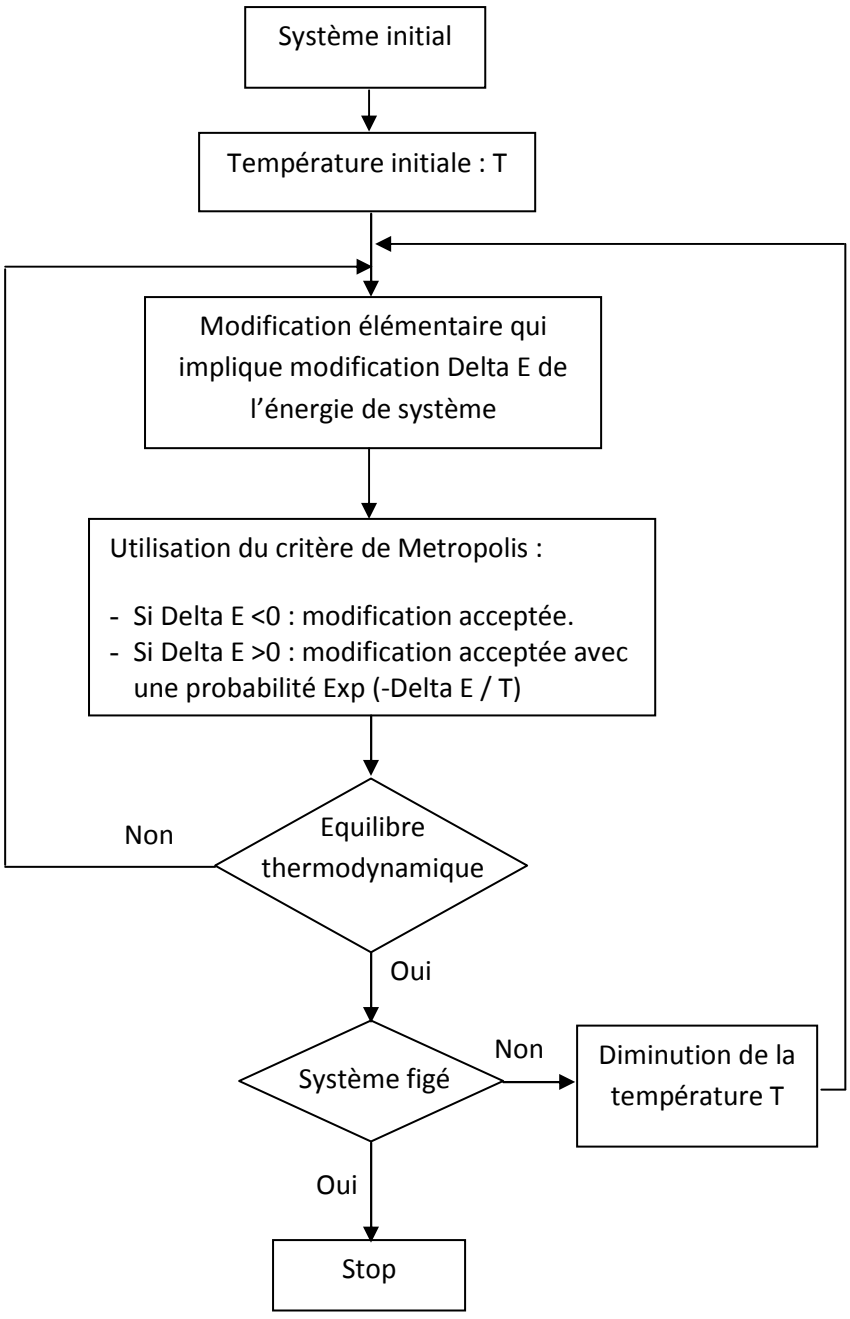


Figure III.1 : L'organigramme de l'algorithme du recuit simulé

III.2.2 Les grandes lignes de l'algorithme de recuit simulé :

Algorithme du recuit simulé (T°, N);

***T°** : température initiale*
***N** : nombre d'itérations par palier*

Début

S := Solution initiale; // initialiser aléatoirement de coût E_S.
T := T°;
Tant que T >0 faire
début
J := 0;
Tant que J < N faire // Une étape
début
S' := Générer une transition à partir de S; // par changement d'un bit choisi aléatoirement de S.
$\Delta E := E_S - E_{S'}$;
Si $\Delta E \leq 0$ Alors S := S' // Accepter la transition
Sinon début
Proba := exp (– $\Delta E/T$)
Si aléatoire (0,1) <= Proba Alors S := S' // Accepter la solution
fin;
J := J+1;
fin;
T := T-1;
fin;
Solution finale := S;

Fin.

Figure III.2 : Les grandes lignes de l'algorithme de la technique de recuit simulé.

III.3 SVM (Machines à Vecteurs de Support) :

Parmi les méthodes à noyaux, inspirées de la théorie statistique de l'apprentissage de Vladimir Vapnik, les SVM constituent la forme la plus connue. SVM est une méthode de classification binaire par apprentissage supervisé, elle fut introduite par Vapnik en 1995. Cette méthode est donc une alternative récente pour la classification. Cette méthode repose sur

l'existence d'un classificateur linéaire dans un espace approprié. Puisque c'est un problème de classification à deux classes, cette méthode fait appel à un jeu de données d'apprentissage pour apprendre les paramètres du modèle. Elle est basée sur l'utilisation de fonction dites noyau (kernel) qui permettent une séparation optimale des données.

Dans la présentation des principes de fonctionnements, nous schématiserons les données par des « points » dans un plan.

III.3.1 Principe de fonctionnement général de SVM :

III.3.1.1 Notions de base : Hyperplan, marge et support vecteur

Pour deux classes d'exemples donnés, le but de SVM est de trouver un classificateur qui va séparer les données et maximiser la distance entre ces deux classes. Avec SVM, ce classificateur est un classificateur linéaire appelé hyperplan.

Dans le schéma qui suit, on détermine un hyperplan qui sépare les deux ensembles de points.

Les points les plus proches, qui seuls sont utilisés pour la détermination de l'hyperplan, sont appelés vecteurs de support.

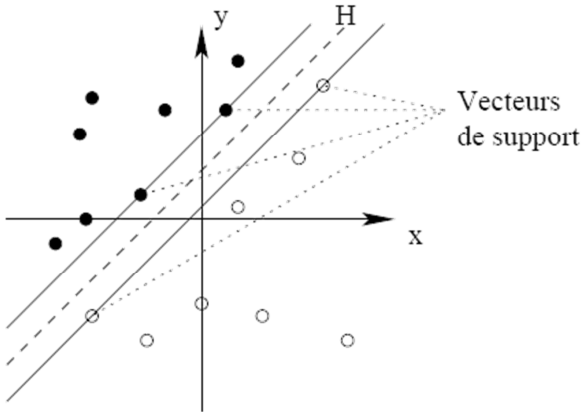


Figure III.3 : un hyperplan

Il est évident qu'il existe une multitude d'hyperplan valide mais la propriété remarquable des SVM est que cet hyperplan doit être optimal. Nous allons donc en plus chercher parmi les hyperplans valides, celui qui passe « au milieu » des points des deux classes d'exemples. Intuitivement, cela revient à chercher l'hyperplan le « plus sûr ». En effet, supposons qu'un exemple n'ait pas été décrit parfaitement, une petite variation ne modifiera

pas sa classification si sa distance à l'hyperplan est grande. Formellement, cela revient à chercher un hyperplan dont la distance minimale aux exemples d'apprentissage est maximale. On appelle cette distance « marge » entre l'hyperplan et les exemples. L'hyperplan séparateur optimal est celui qui maximise la marge. Comme on cherche à maximiser cette marge, on parlera de séparateurs à vaste marge.

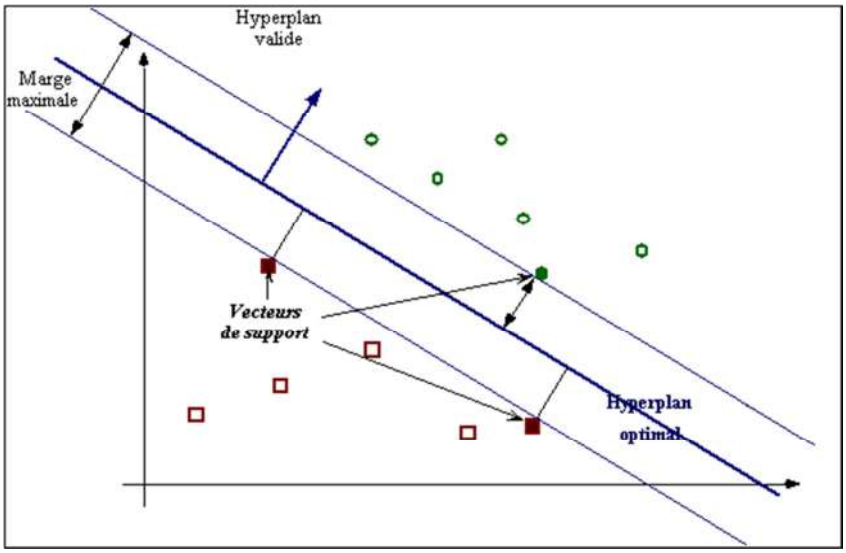


Figure III.4 : L'hyperplan séparateur optimal

III.3.1.2 Pourquoi maximiser la marge ?

Intuitivement, le fait d'avoir une marge plus large procure plus de sécurité lorsque l'on classe un nouvel exemple. De plus, si l'on trouve le classificateur qui se comporte le mieux vis-à-vis des données d'apprentissage, il est clair qu'il sera aussi celui qui permettra au mieux de classer les nouveaux exemples. Dans le schéma qui suit, la partie droite nous montre qu'avec un hyperplan optimal, un nouvel exemple reste bien classé alors qu'il tombe dans la marge. On constate sur la partie gauche qu'avec une plus petite marge, l'exemple se voit mal classé.

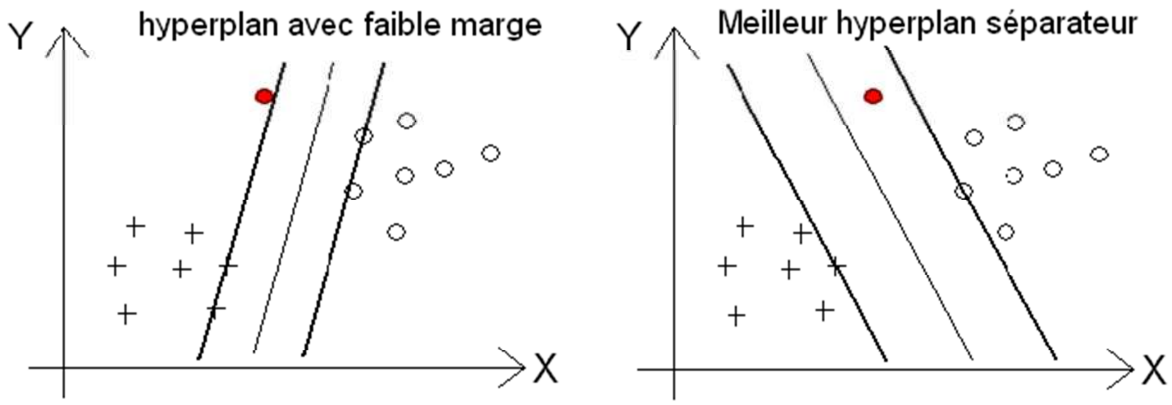


Figure III.5 : hyperplan optimal et hyperplan avec faible marge.

En général, la classification d'un nouvel exemple inconnu est donnée par sa position par rapport à l'hyperplan optimal.

III.3.1.3 Linéarité et non-linéarité

Parmi les modèles des SVM, on constate les cas linéairement séparable et les cas non linéairement séparable. Les premiers sont les plus simples de SVM car ils permettent de trouver facilement le classificateur linéaire. Dans la plupart des problèmes réels il n'y a pas de séparation linéaire possible entre les données, le classificateur de marge maximale ne peut pas être utilisé car il fonctionne seulement si les classes de données d'apprentissage sont linéairement séparables.

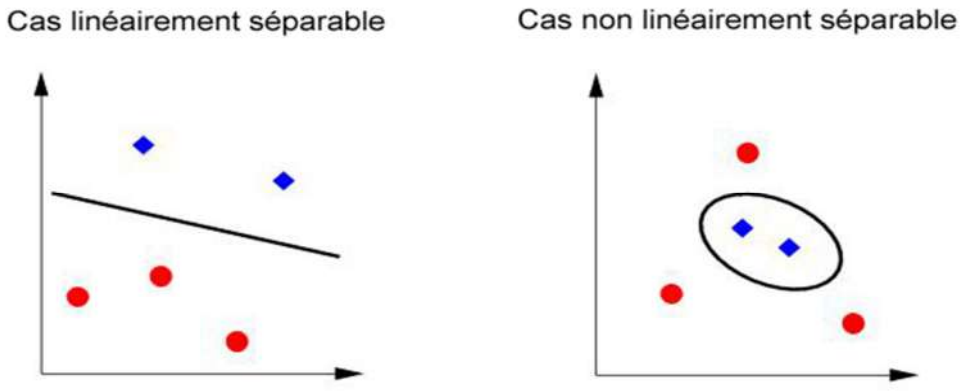


Figure III.6 : Linéarité et non-linéarité

III.3.1.3.1 Cas non linéaire

Pour surmonter les inconvénients des cas non linéairement séparable, l'idée des SVM est de changer l'espace des données. La transformation non linéaire des données peut permettre une séparation linéaire des exemples dans un nouvel espace. On va donc avoir un changement de dimension. Cette nouvelle dimension est appelé « espace de re-description ». En effet, intuitivement, plus la dimension de l'espace de re-description est grande, plus la probabilité de pouvoir trouver un hyperplan séparateur entre les exemples est élevée. Ceci est illustré par le schéma suivant :

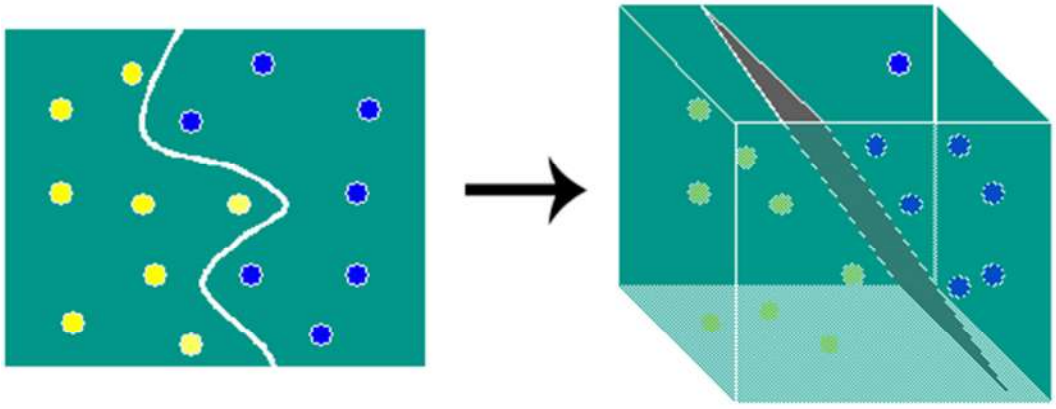


Figure III.7 : Cas non linéaire vers cas linéaire

On a donc une transformation d'un problème de séparation non linéaire dans l'espace de représentation en un problème de séparation linéaire dans un espace de re-description de plus grande dimension. Cette transformation non linéaire est réalisée *via* une fonction noyau. En pratique, quelques familles de fonctions noyau paramétrables sont connues et il revient à l'utilisateur de SVM d'effectuer des test pour déterminer celle qui convient le mieux pour son application. On peut citer les exemples de noyaux suivants : polynomiale, gaussien, sigmoïde et laplacien.

III.3.1.3.2 Illustration de transformation de cas non linéaire : le cas XOR

Le cas de XOR n'est pas linéairement séparable, si on place les points dans un plan à deux dimension, on obtient la figure suivante

Coordonnées des points : (0,0) ; (0,1) ; (1,0) ; (1,1)

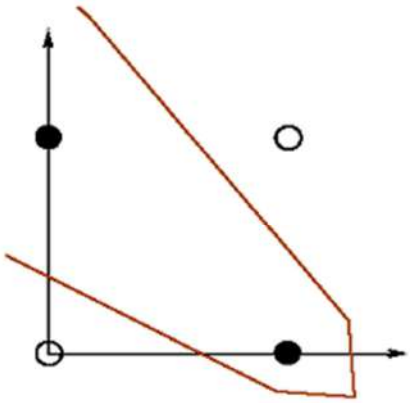


Figure III.8 : Le cas XOR (dimension 2)

Si on prend une fonction polynomiale (x , y) -> (x , y , x.y) qui fait passer d'un espace de dimension 2 à un espace de dimension 3, on obtient un problème en trois dimensions linéairement séparable [8] :

(0,0) -> (0,0,0) , (0,1) -> (0,1,0) , (1,0) -> (1,0,0) , (1,1) ->(1,1,1)

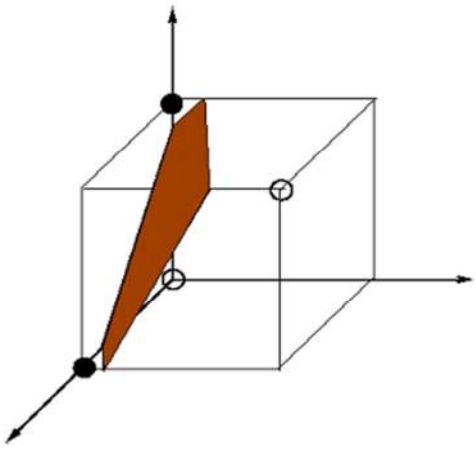


Figure III.9 : Le cas XOR (dimension 3)

III.3.2 SVM multi classe :

La plupart des problèmes ne se contentent pas de deux classes de données. Il existe plusieurs méthodes pour faire la classification multiclasses. Citons les plus utilisées : La première méthode, celle que nous utilisons dans notre application, est une méthode dite Un-contre-Un. Au lieu d'apprendre N fonctions de décisions, ici chaque classe est discriminée d'une autre. La deuxième méthode est appelé Un-Contre-Tous. C'est une approche étendant la notion de

marge aux cas multiclasses. Cette formulation intéressante permet de poser un problème d'optimisation unique. Le problème fait intervenir N fonctions de décision.

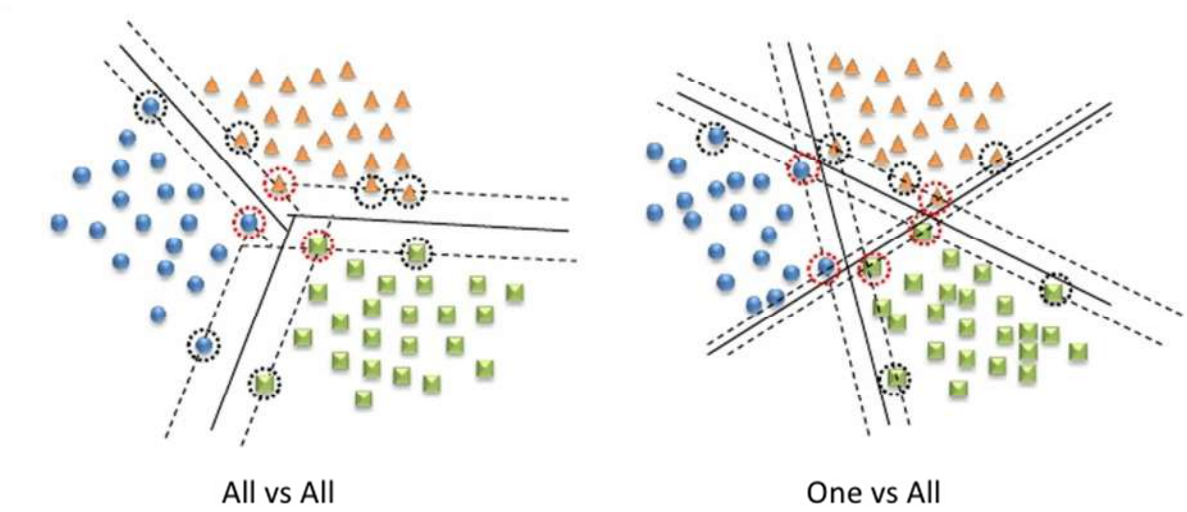


Figure III.10 : Exemples SVM multi-classes

III.4 Méthode d'enveloppe proposée :

La méthode d'enveloppe que nous présentons est une approche de type intégrée pour faire face au problème de sélection de gènes pour la classification des données de puces à ADN. Notre approche sélectionne un sous-ensemble de gènes grâce à la méthode de recuit simulé. Dans cette méthode, la qualité d'un sous-ensemble de gènes est évaluée grâce à un classificateur SVM construit sur ce sous-ensemble. De plus, une fonction d'évaluation de nos solutions (sous-ensembles de gènes) prend en compte le taux de classification pour évaluation d'un sous-ensemble.

III.4.1 Organigramme :

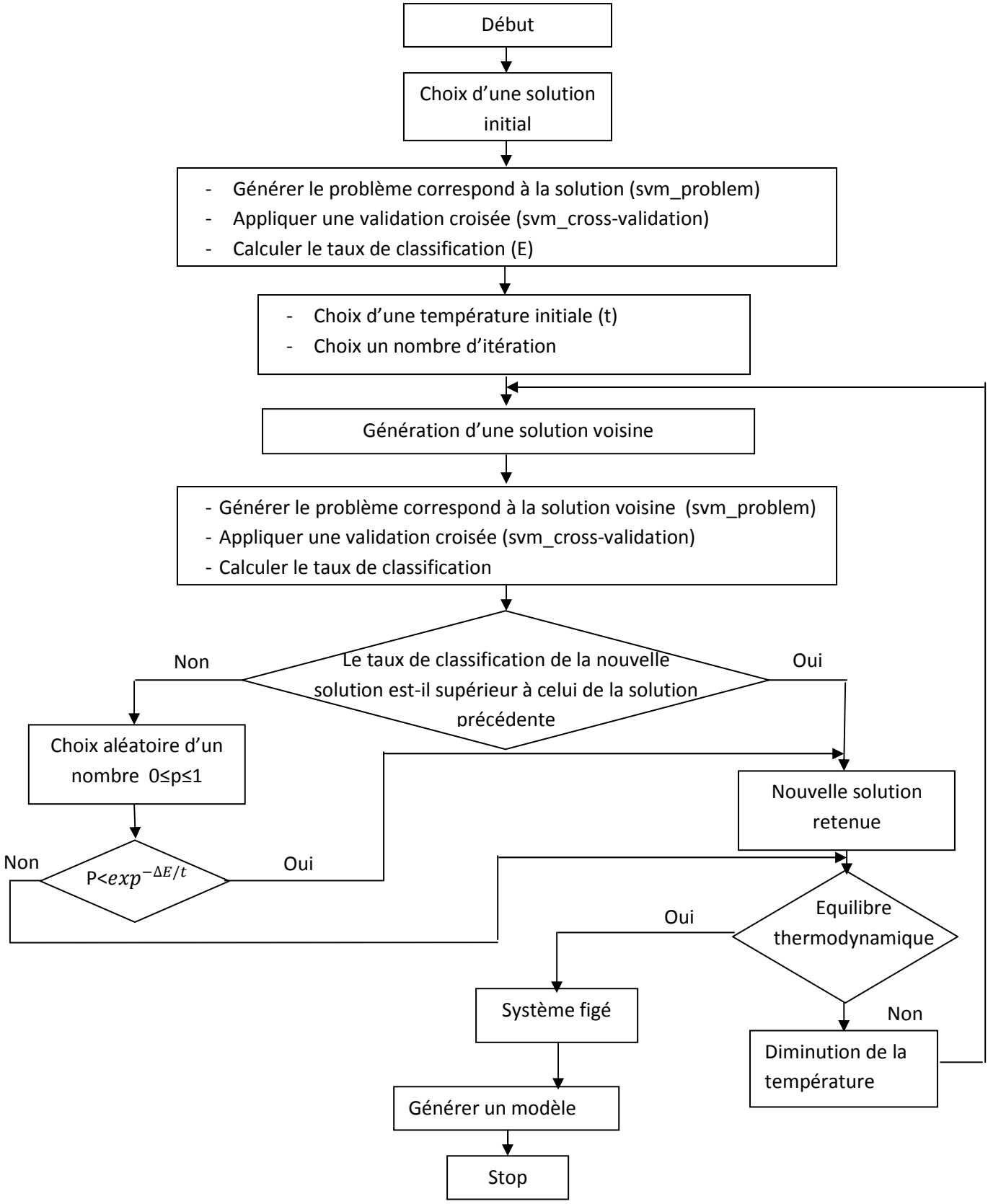


Figure III.11 : Organigramme de la méthode proposée

III.4.2 Architecture :

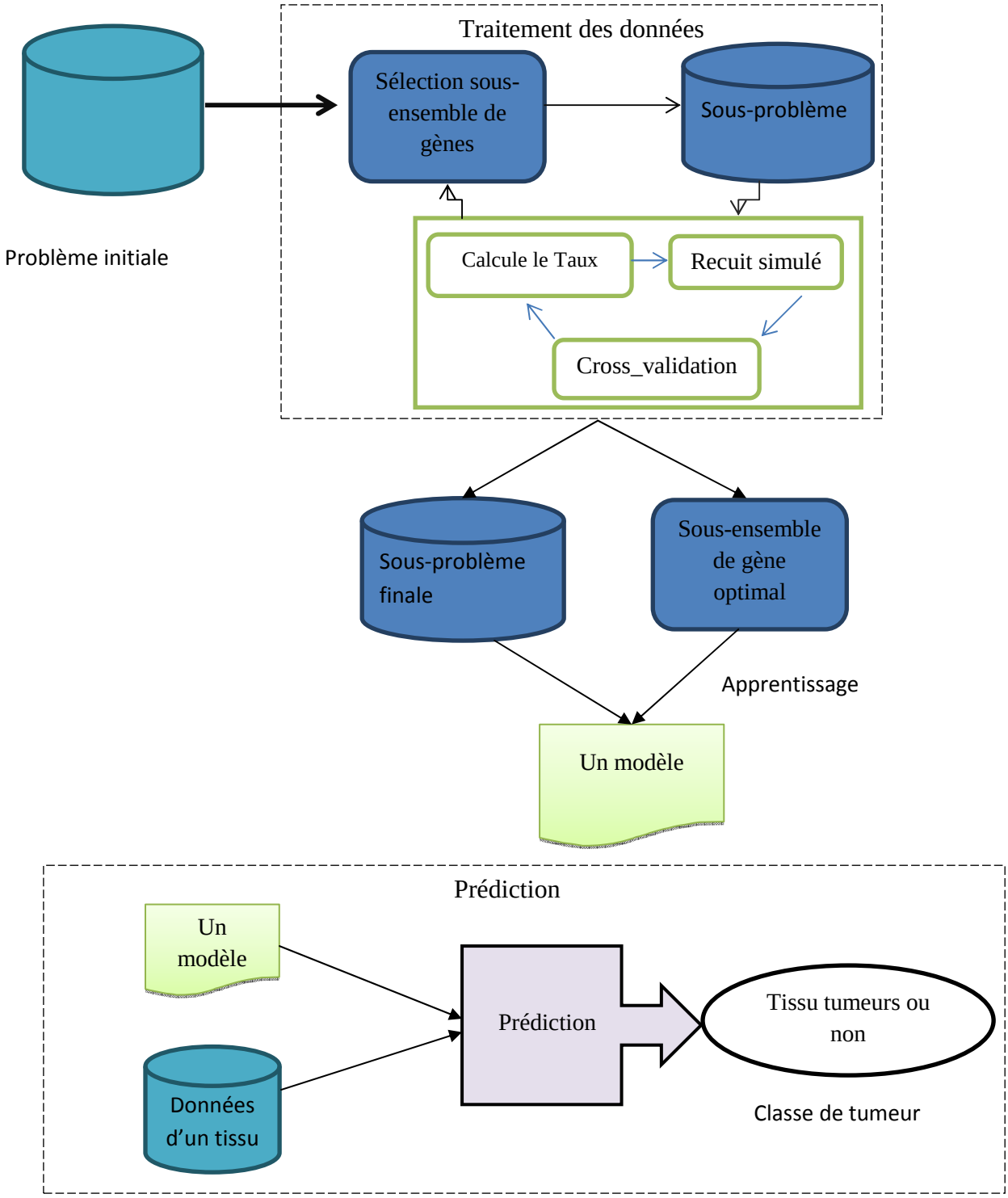


Figure III.12 : Architecteur de développement.

III.5 Conclusion :

Dans ce chapitre, nous avons proposé une nouvelle approche de sélection de gènes à ADN basée sur le recuit simulé utilisant SVM pour l'évaluation des sous-ensembles candidats. Mais pour l'implémentation on a besoin d'une étude conceptuelle de cette méthode. Cette étape nous donne une vision globale sur la méthode proposée.

CHAPITRE IV

Conception du système de prédiction

IV.1 Introduction :

L'application à réaliser s'inscrit dans le cadre de la prédiction , qui se déroule principalement en deux étapes : une phase d'apprentissage et une phase de test, Le prédicteur reçoit en entrée des données d'apprentissage (dans notre cas des données issues des bio-puces) qu'il utilise pour générer un modèle, il reçoit par la suite un fichier de test, dont il se servira pour prédire la présence de certaine maladie, en particulier les cancers (différencier les tissus tumoraux et les tissus sains) en se basant sur le modèle préalablement généré.

IV.2 Fonctionnalités du système :

Le système doit permettre à l'utilisateur d'effectuer un certain nombre de tâches, qui ne sont autre que les étapes du processus de prédiction antérieurement décrits, à rappeler : définir des paramètres d'entrée, charger des données d'apprentissage pour générer un modèle, puis utiliser ce modèle pour faire des prédictions. Les fonctionnalités utilisateurs sont rassemblées dans le diagramme des cas d'utilisation suivant.

IV.2.1 Diagramme des cas d'utilisation :

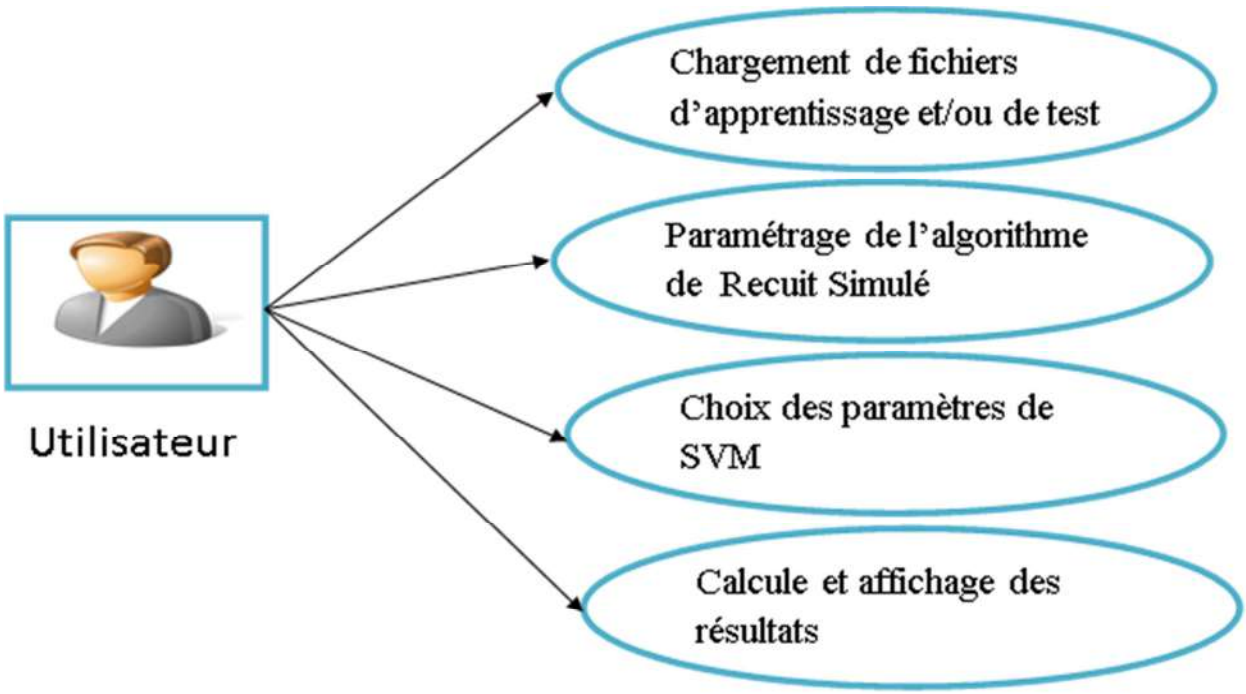


Figure IV.1 : Diagramme de cas d'utilisation

IV.2.2 Description des cas d'utilisation :

Ce diagramme est assez simple et ne comporte qu'un acteur. Au niveau des cas d'utilisation, l'utilisateur peut charger le fichier de données, régler et choisir les paramètres du recuit simulé et même SVM. Il peut également calculer le taux de classification qui indique la qualité des données, et enfin afficher les résultats.

A partir de ce diagramme, nous avons défini les différentes étapes pour évaluer les données, et nous avons établi un diagramme d'activité (FigureIV.2), indiquant le chargement des données d'apprentissage.

IV.2.3 Diagrammes d'activités :

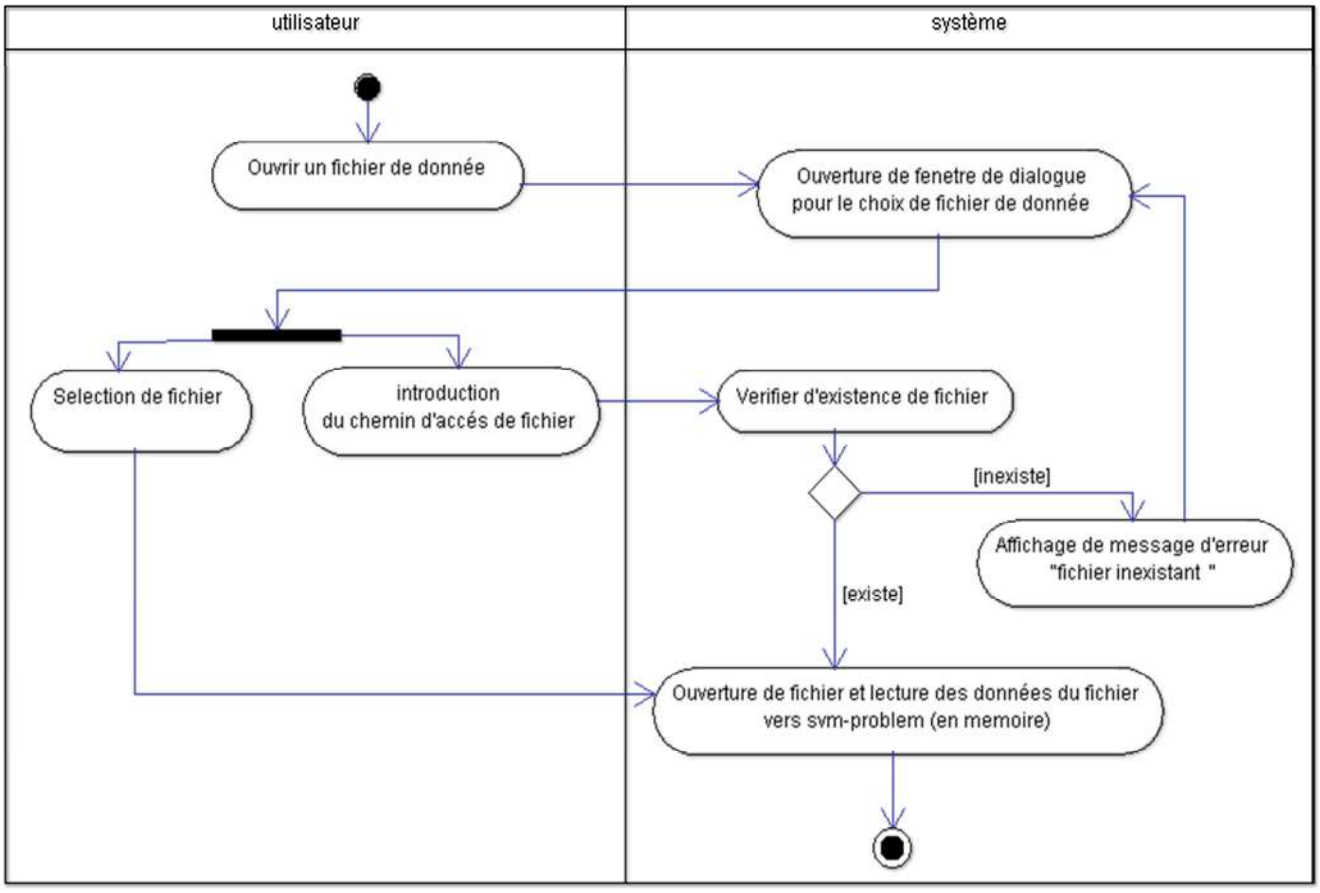


Figure IV.2 : Chargement des données d'apprentissage

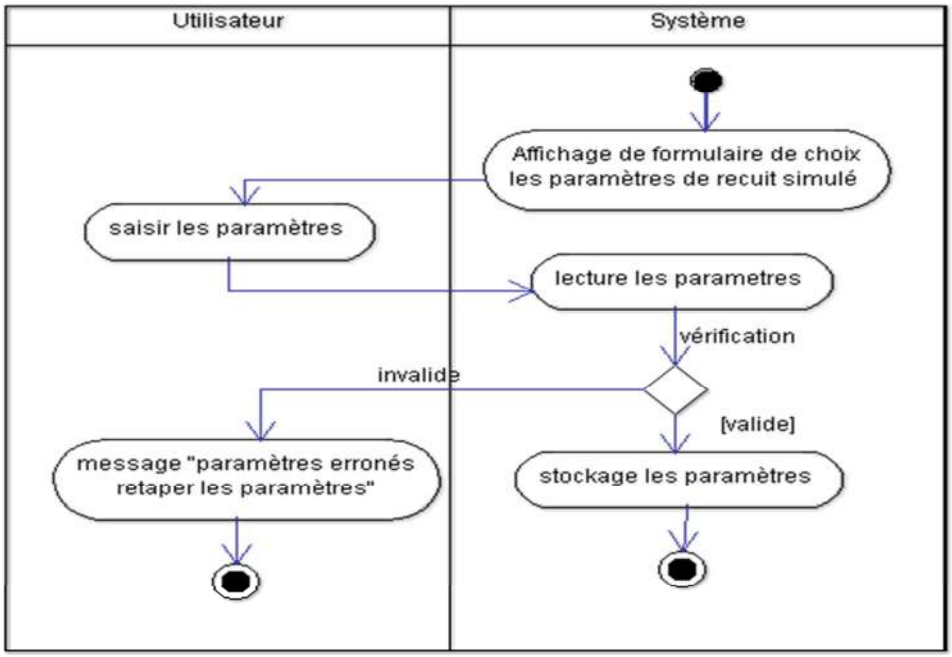


Figure IV.3 : Paramétrage de l'algorithme de recuit simulé

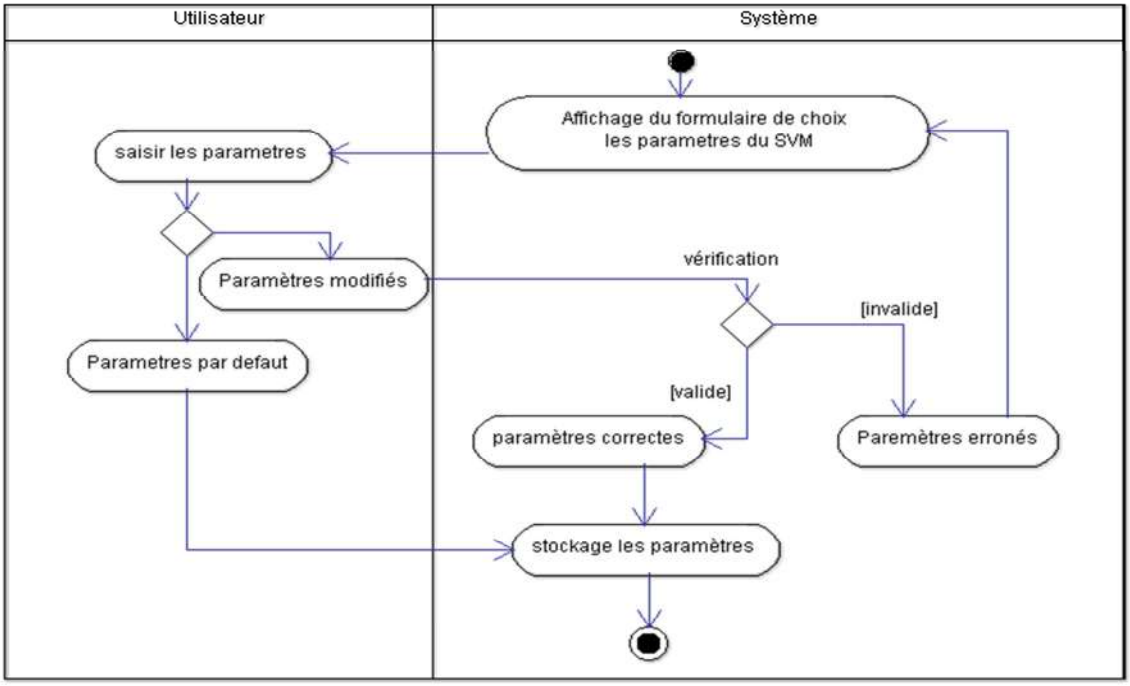


Figure IV.4 : Choix des paramètres du SVM

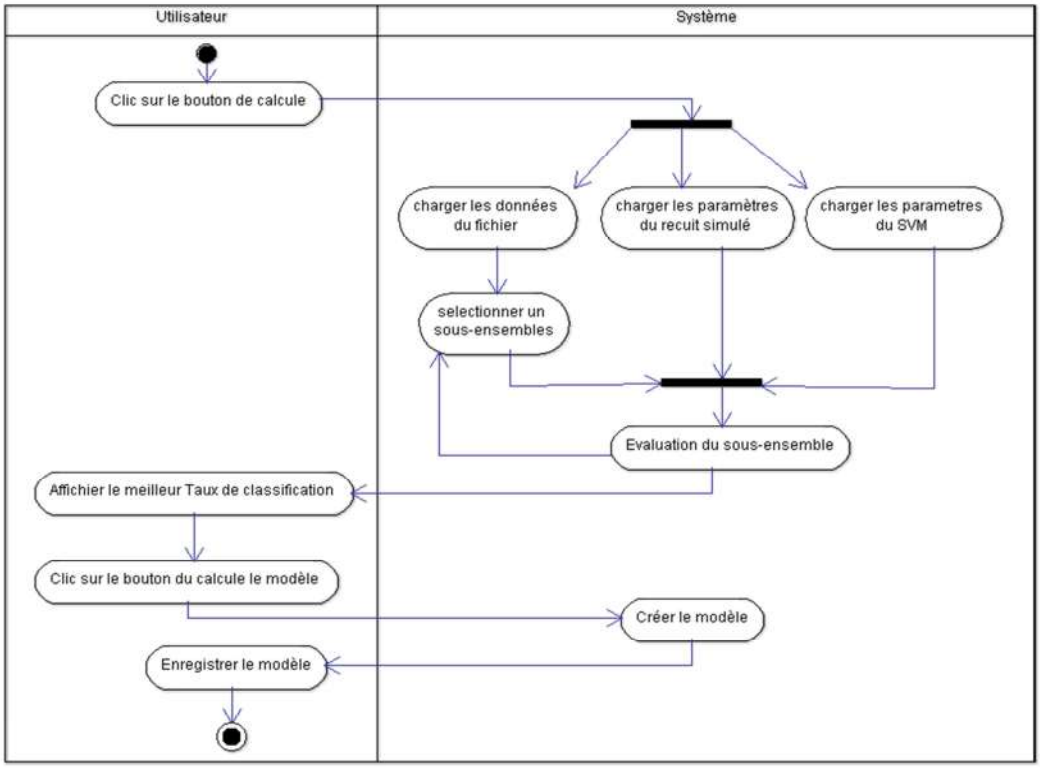


Figure IV.5 : Calcul et affichage de résultats

IV.3 Description des outils utilisés :

IV.3.1 LIBSVM :

La LIBSVM est une librairie dédiée aux Machines à Support de Vecteurs (Support Vector Machines) élaborée par Chih-Chung Chang et Chih-Jen Lin, deux chercheurs du département informatique de l'université de Taipei à Taiwan « National Taiwan University ». La librairie englobe un ensemble de fonctions implémentant des algorithmes de fouille de données, particulièrement utilisée pour mettre en œuvre des classifications et des régressions. Elle est disponible en langage C++ et en JAVA, et compatible avec diverses plateformes logicielles (Python, R, MATLAB, Perl, Ruby,Weka, Common LISP, CLISP, Haskell, LabVIEW, interfaces PHP, C# .NET, extensions CUDA).

La LIBSVM est vastement exploitée dans le domaine de l'apprentissage automatique (learning machines), principalement en supervisé. Elle a gagné en popularité depuis le début des années 2000 et jusqu'à maintenant vu qu'elle traite divers problèmes d'optimisation (minimisation) en intégrant des solveurs pour la classification de type C-support vector (C-

SVC) et ν-support vector (ν-SVC), pour l'estimation de la distribution (one-class SVM), pour la régression de type ϵ-support vector (ϵ-SVR), et ν-support vector (ν-SVR), ainsi que pour la classification multi-classes. Le temps de traitement sont particulièrement satisfaisant tout autant que les résultats obtenus, raison pour laquelle la librairie est considérée comme référence en la matière.

Un site web dédié à la LIBSVM a été dressé par les auteurs pour fournir la documentation et les guides d'utilisation de la librairie, les exécutables et les codes sources en plusieurs langages (un point fort de l'outil), des jeux de données pour les tests, ainsi que les dernières mises à jours : http://www.csie.ntu.edu.tw/~cjlin/libsvm. De même une foire aux questions est présente sur le même site etest significativement constructive. Il est possible de contacter les auteurs pour toute remarque, report de bug ou suggestion d'amélioration de la librairie.

IV.3.1.1 Fonctionnalités de LIBSVM :

La LIBSVM peut être exploitée en l'intégrant à un code applicatif donné, ou directement en faisant appel aux fichiers exécutables. Que l'utilisateur tourne sous un environnement Windows ou Unix, il peut facilement récupérer les makefile ou les DLL disponibles en ligne, puis de les compiler pour générer leurs .exe respectifs. Après compilation, l'utilisateur peut faire appel aux fonctionnalités de la librairie en ligne de commande, ci-dessous un descriptif des commandes les plus basiques :

- Normalisation des données : svm-scale : La normalisation des données est préconisée dans la plupart des traitements d'analyse, la LIBSVM appuie cette pratique et propose une fonctionnalité pour la normalisation: svm-scale, elle recommande une normalisation linéaire utilisant les intervalles [-1; +1] ou [0; 1], en utilisant bien sûr la même intervalle pour le jeu de données d'apprentissage et de test.
- Apprentissage : svm-train : La commande svm-train permet de lancer le processus d'apprentissage sur les données, elle prend en entrée un format de données spécifique, décrit plus loin, et génère un fichier de modèle.
- Prédiction : svm-predict : Génère les classes prédites ou les valeurs prédites (en cas de régression), en se basant sur un modèle entrainé, la commande à utiliser est la suivante : svmpredicttest_filemodel_fileoutput_file

  Où test_file est fichier de test, ou les données que l'on souhaite prédire, à noter qu'il doit également être sous le format LIBSVM et model_file le nom du modèle à générer.

Après prédiction, la fonction svm-predict compare les résultats de la prédiction avec les labels du fichier de test, pour en déduire la justesse de la prédiction et par la même occasion le taux d'erreur commis.

IV.3.1.2 Paramètres de SVM :

Afin d'entamer le processus d'apprentissage avec la LIBSVM, certains paramètres sont à renseigner selon qu'on souhaite faire une classification, une régression ou autre, le choix des bons paramètres est déterminant pour obtenir des résultats satisfaisants. Ci-dessous la liste de tous les paramètres LIBSVM possibles. À noter que dans le cas où ces paramètres ne sont pas renseignés, la librairie utilise les valeurs par défaut, précisées ci-après également :

-s svm_type : C'est le type de l'algorithme SVM à utiliser, peut être l'une des fonctions: C_SVC,NU_SVC, ONE_CLASS, EPSILON_SVR, NU_SVR, et peut prendre respectivement les valeurs suivantes :

0: Pour une classification de type C-SVM

1: Pour une classification de type nu-SVM

2: Pour une classification de type one-class-SVM

3: Régression de type epsilon-SVM

4: Régression de type nu-SVM

Par défaut le type SVM utilisé est C_SVC.

-t kernel_type : C'est le type de la fonction noyau à utiliser, peut être défini à : LINEAR, POLY, RBF ou SIGMOID, et par conséquent prendre les valeurs :

0: Pour utiliser une fonction linéaire de formule =u'*v

1: Pour une fonction polynomiale de formule = (gamma*u'*v + coef0)^degré

2: Pour une fonction Radiale de formule = exp(-gamma*|u-v|^2)

3: Pour une fonction sigmoïde de formule = tanh(gamma*u'*v + coef0)

4: Pour créer un kernel dans le fichier de test (cf. documentation LIBSVM pour le détail).

Telsque : u' représente la transposé du vecteur contenant les valeurs des attributs de l'ensemble d'apprentissage, et v le vecteur des labels (étiquettes). Le gamma, degré et coef0 sont des paramètres (rentrés pas l'utilisateur). Par défaut le type de la fonction noyau utilisée est RBF, la documentation LIBSVM recommande l'utilisation de la fonction RBF pour plusieurs raisons, entre autre parce que RBF gèrele cas où la relation entre les labels et les attributs est non linéaire.

Paramètres des fonctions noyau :

-*d* `degree` : Paramètre degré de la fonction noyau, par défaut 3

-*g gamma* : Paramètre gamma de la fonction noyau, par défaut 1

-*r coef0* : Paramètre coef0 de la fonction noyau, par défaut 0.

Paramètres dépendants du type SVM choisi :

-*c cost* : C'est le paramètre C (coût), qui représente la pénalité de l'erreur, à renseigner lors de l'utilisation du type SVM C-SVC, epsilon-SVR et nu-SVR, par défaut le coût est égal à 1

-*wi weight* : pour changer le paramètre C à weight*C, s'il n'est pas renseigné weight est égale à 1 sa valeur par défaut, et par conséquent neutre.

-*n nu* : Paramètre nu du type nu-SVC, One-class-SVM et nu-SVR, par défaut 0.5

-*p epsilon*: Paramètre epsilon de la fonction de perte (LossFunction) pour le type epsilon-SVR, par défaut égal à 0.1

Paramètres SVM généraux :

-*m cachesize* : Pour paramétrer la taille mémoire (Mbits) allouée au noyau, par défaut 100

-*e epsilon* : C'est le critère d'arrêt de l'apprenant, par défaut 0.001. Il est recommandé d'utiliser une valeur de 0.00001 pour le nu-SVC, et 0.001 pour les autres types.

-*h shrinking* : Pour activer/désactiver l'heuristique de shrinking, par défaut mis à 1.

-*b probability-estimates* : Pour activer/désactiver l'apprentissage avec probabilité, par défaut à 0.

-*v n* : paramètre spécifique au mode validation-croisée (cross-validation), le n est le nombre de sous-divisions de l'ensemble d'apprentissage (n-fold), par défaut désactivé.

-*q*: Mode silencieux, pour omettre l'affichage des messages, par défaut mis à 0.

À noter que la librairie propose un utilitaire pour vérifier la validité des paramètres avant de faire appel à la fonction d'apprentissage, vie la commande svm_check_parameter().

IV.3.1.3 Structure des fichiers de données :

-**Entrées** : **Fichier au format LIBSVM**

Les fonctions de la LIBSVM reçoivent en paramètre des fichiers textes, qui doivent suivre un format bien particulier, il s'agit en fait d'un tableau où les données doivent commencer par le libellé de la classe (habituellement 1 ou -1), suivi d'une série de données sous la forme de paires (index :valeur).

En ouvrant un fichier d'apprentissage ou un fichier de test on retrouve l'ensemble du jeu de données sous la forme:

[label] [index1]:[value1] [index2]:[value2] ...

[label] [index1]:[value1] [index2]:[value2] …

[label] [index1]:[value1] [index2]:[value2] ...

Label: C'est le libellé ou la classe de l'enregistrement. C'est un entier lors d'une classification, et un réel lors d'une régression.

Index : Indexes dans l'ordre croissant, ils sont généralement contiguës.

Value : Ce sont les valeurs à apprendre, généralement des nombres réels.

Chaque ligne du fichier représente un enregistrement(ou un individu) appartenant un une classe du jeu de données, comme sur l'exemple ci-dessous :

+1 1:0.708 2:1 3:1 4:-0.320 5:-0.105 6:-1

- **Fichier modèle :**

Le fichier du modèle contient les paramètres qui ont été utilisés lors de la phase d'apprentissage suivis d'un tableau dont les lignes représentent les supports vecteurs. Les supports vecteurs listés dans l'ordre des labels, et regroupés par classes. Sachant que pour un jeu de données de k classes, on aura k-1 coefficients pour les supports vecteurs de chacune des classes.

- **Fichier de prédiction**

Le format du fichier de prédiction résultant après l'appel de la fonction svm-predict est simple, chaque ligne de ce fichier contient le label relatif à la classe prédite pour la ligne équivalente du fichier de test.

IV.3.2 Builder C++ XE4 :

C++Builder est un logiciel de développement rapide d'applications (abr. RAD) conçu par Borland qui reprend les mêmes concepts, la même interface et la même bibliothèque que Delphi en utilisant le langage C++. Il permet de créer rapidement des applications Windows ainsi qu'une interface graphique avec son éditeur de ressources. Il est compatible avec la norme ISO C++.

La version XE4 d'Embarcadero® est une solution pour de développement multi-système permettant de créer des applications véritablement natives et à hautes performances pour PC, tablettes et smartphones.

IV.4 Structure technique du système :

L'application à réaliser est simple d'un point de vue complexité des composants, il est toutefois important de la coder de façon modulaire, et de réaliser une séparation entre les couches logiques qui la composent, et ce pour une meilleure lisibilité du code et une meilleure standardisation de l'architecture logicielle. La séparation des entités logiques permet de concevoir des applications standards extensibles et surtout faciles à maintenir et à faire évoluer. On peut par exemple aisément changer la manière d'accéder aux données sans se

soucier de la partie présentation, ou de la manière avec laquelle ces données seront manipulées. L'application sera donc découpée en 3 couches.

IV.4.1 Couches du système :

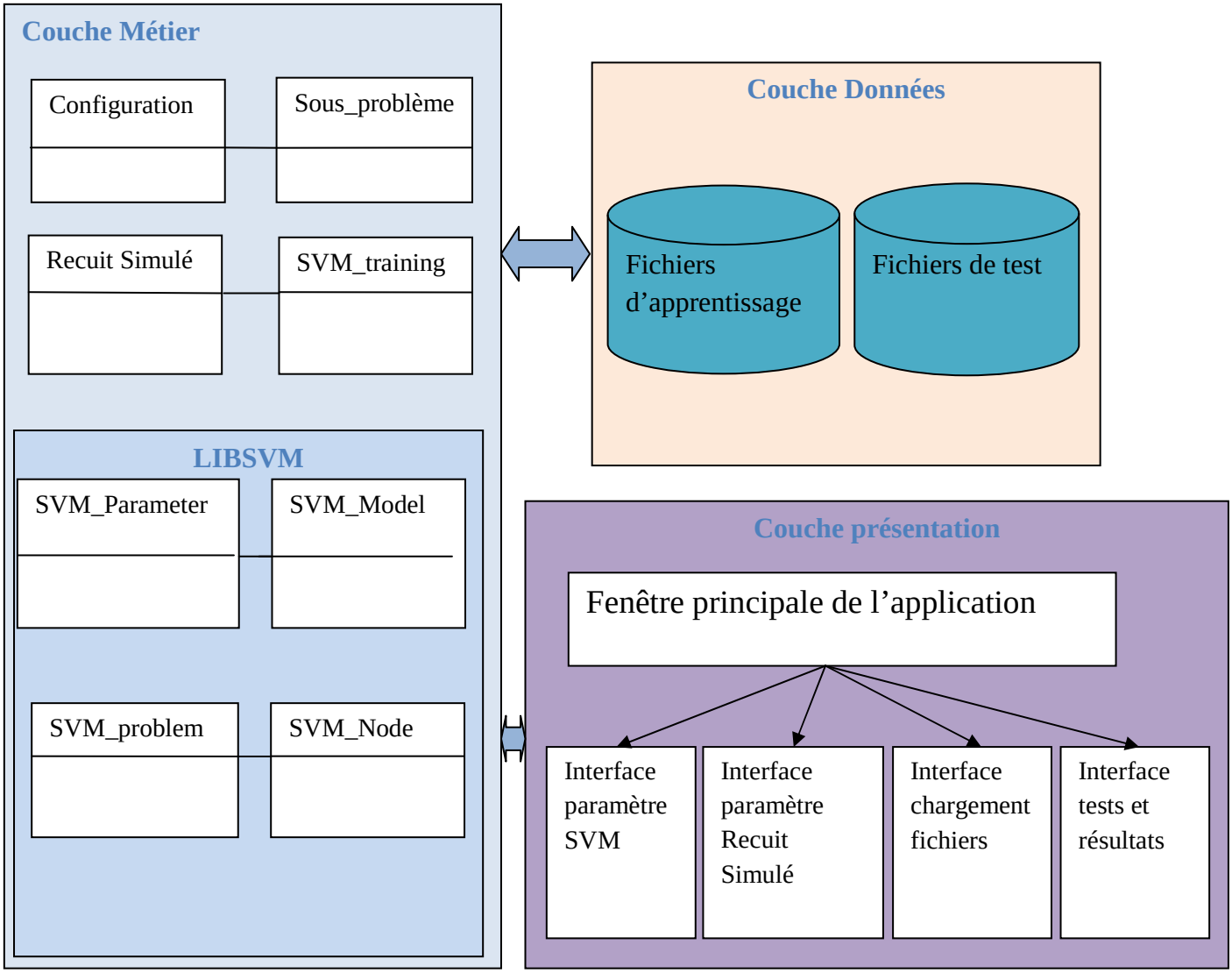


Figure IV.3 : Découpage de l'application de prédiction en 3 couches logiques.

- **Couche Métier :**

La couche métier regroupe toutes les données et méthodes qui décrivent la logique du processus de prédiction (chargement des fichiers, apprentissage des données, sélection de gènes par Recuit Simulé, prédiction…)

- **Couche Données :**

Représente les données de l'application, sachant qu'elles sont stockées dans des fichiers textes (base de données d'apprentissage et/ou de test).

- **Couche Présentation :**

Rassemble le Contrôleur de l'application (fenêtre principale) ainsi que les interfaces homme-machine, c'est l'interface vue par l'utilisateur.

IV.4.2 Diagramme de classes :

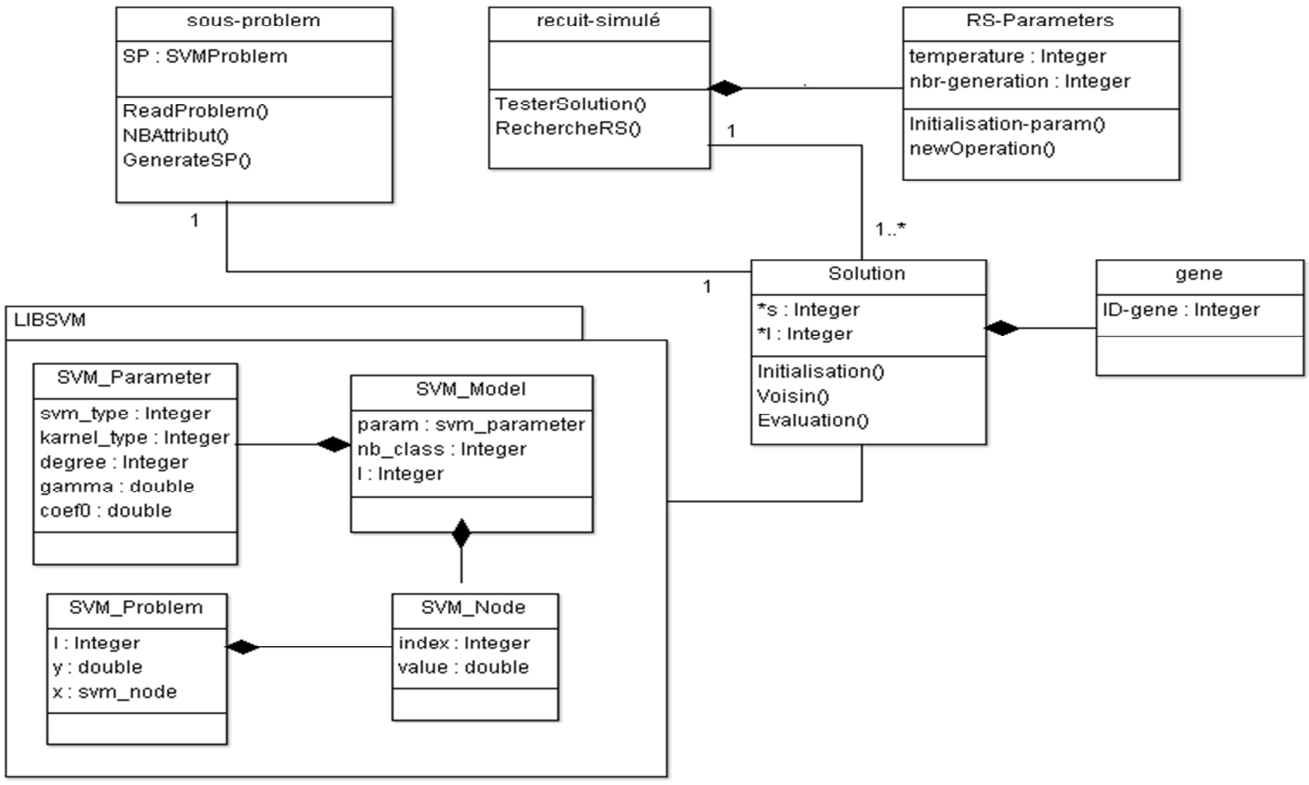


Figure IV.4 : Diagramme de classes.

IV.5 Conclusion :

Nous avons présenté, dans ce chapitre les diagrammes de cas d'utilisation, de classe et les diagrammes d'activités. Au cours de cette analyse nous avons donné une vue générale sur notre méthode.

CHAPITRE V

Implémentation et résultats

V.1 Introduction :

Ce chapitre a pour objectif majeur de présenter la réalisation de notre application. Il est composé de deux parties : la première partie présente l'environnement de développement alors que la seconde partie concerne la présentation de l'interface graphique et les résultats expérimentaux.

V.2 Aperçu de l'application développée :

L'ergonomie de l'interface graphique de l'application n'a pas été particulièrement étudiée, l'intérêt étant plus centré sur le développement de la méthode de sélection des gènes et l'obtention des résultats de classification probants .C'est une interface simple avec une organisation des plus basiques, elle contient des commandes standards communes : boutons, case à cochée, listes déroulantes, zones de texte…, il pourra être amélioré par la suite.

Au démarrage de l'application, La fenêtre principale s'affiche, elle est composée de 4 parties :

Chargement des données (1), choix de paramètres de Recuit simulé (2), choix de paramètres de SVM (3), en plus d'une zone pour lancement d'un test et l'affichage des résultats (4).

Chargement des données : L'utilisateur commence par renseigner le chemin des fichiers de données d'apprentissage.

Choix de paramètres de Recuit simulé : L'utilisateur renseigne par la suite les paramètres de l'algorithme de recuit simulé :

- ***T°* :** température initiale.
- ***N*:** nombre d'itérations par palier

Choix de paramètres de SVM : L'utilisateur renseigne à ce niveau les paramètres des deux phases d'apprentissage et de prédiction : Type du SVM, Type du noyau, coût, Epsilon, Nu, Gamma, Degré, Taille du cache utilisée …

(Voir la section 4.3.1 du chapitre 4 pour la signification des paramètres SVM et le rôle de la sélection de paramètres dans le processus d'apprentissage).

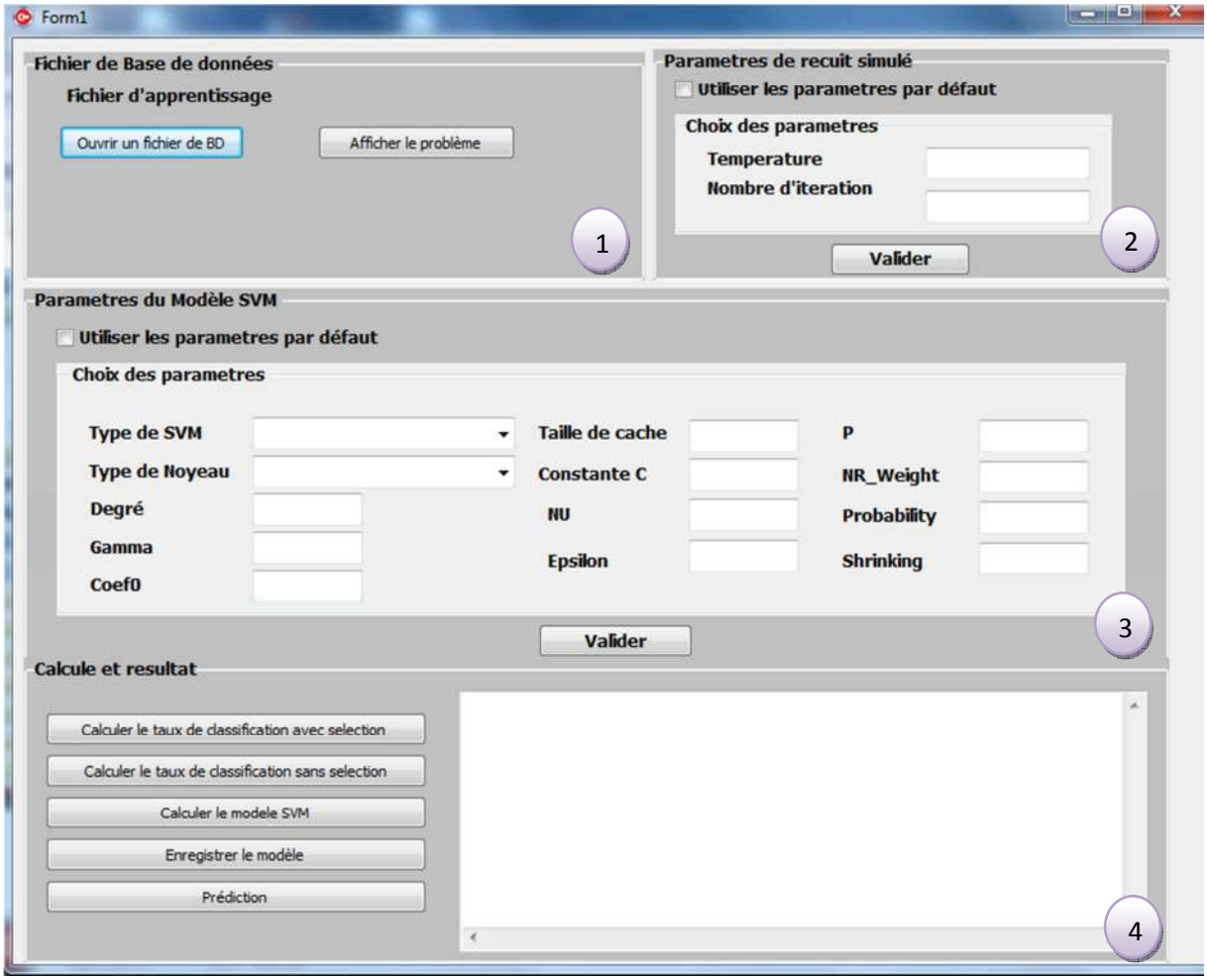


Figure V.1 : Interface de l'application

Test et affichage des résultats : Après l'exécution du processus d'apprentissage, la zone des résultats affiche un message pour en signaler la fin de calcul, et indique la meilleur solution de sélection des gènes trouvé avec le taux de classification, et permet aussi de créer le fichier model de prédiction à partir de l'ensemble de gènes sélectionnés.

Deux fichiers peuvent être créés comme résultats de calcul :

- Fichier Modèle : Un seul modèle est généré à chaque ensemble de gènes sélectionnés.

- Fichier des prédictions : Les prédictions faites sur la base du fichier test et du modèle sont contenues dans un fichier texte qui donne pour chaque cas la classe prédiquée.

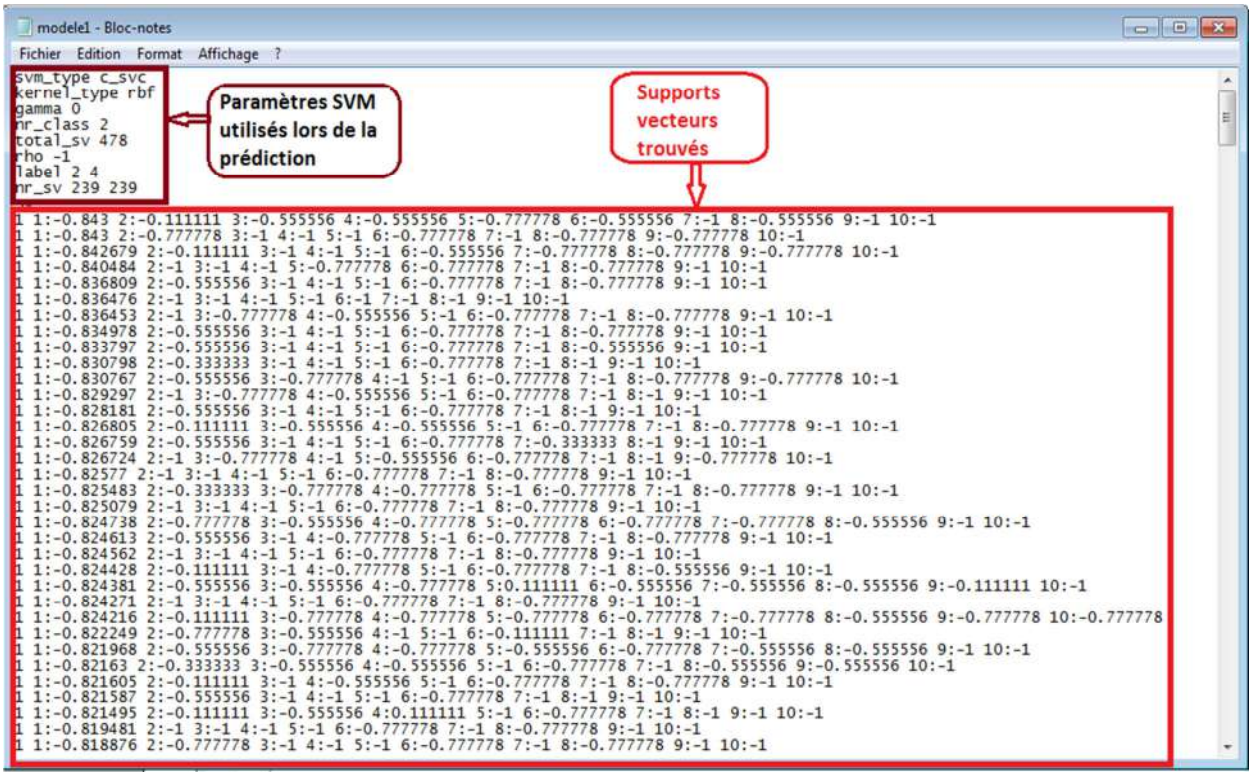


Figure V.2 : Exemple de fichier modèle

V.3 Expériences et résultats :

V.3.1 Présentation des bases de données utilisées :

Pour tester la méthode de sélection de gènes et que nous avons développée, Nous avons utilisé des jeux de données publics, facilement accessibles et qui sont utilisés dans de nombreux travaux concernant la classification. Ces jeux constituent en quelque sorte des jeux tests qui permettent de comparer les méthodes proposées depuis quelques années dans le domaine de la bioinformatique. Nous donnons ci-dessous les caractéristiques de ces jeux de données qui concernent tous des problèmes de reconnaissance de cancers et ou de prévision de diagnostic en oncologie.

V.3.1.1 Cancer du sein

La base de données du cancer du sein dénommée « Wisconsin Breast Cancer Database » a été obtenue par l'Université du Wisconsin. Elle contient les informations médicales de 683 cas cliniques relatifs au cancer du sein classés comme bénin ou malin : 65.5% des patientes sont des cas bénins et 34.5% des cas malins. Les patientes sont caractérisées par 10 attributs.

V.3.1.2 Heart :

La base de données Heart est accessible à l'adresse : http://archive:ics:uci:edu/ml/machine-learning-databases/statlog/heart/heart.dat. Elle et ne contiennent pas des données manquantes. Les données de cette base sont récoltées auprès des patients souffrant des maladies cardiaques. Elles comportent 270 observations et 13 variables.

V.3.1.3 Diabetes

La base Pima Indian Diabetes (PID), disponible sur le site officiel de l'UCI (machine learning repository), offerte par Vincent Sigillito. Cette base est une collection de rapports de diagnostic médicaux de 768 femmes âgé de plus de 21 ans, ces patientes sont des indiennes Pima. Cette base est constitué de 768 cas dont 268 sont diabétique et 500 non diabétique. Chaque cas est formé de 8 attributs.

V.3.2 Prétraitement des données :

Comme dans la plupart des analyses de données, le processus de prédiction avec des machines SVM nécessite la normalisation des données brutes, afin d'éviter que les valeurs à grandeurs extrêmes n'altèrent la qualité de l'analyse. De plus, les fichiers de données d'apprentissage et de test doivent être normalisés sur le même intervalle et de la même manière. La LIBSVM propose une fonctionnalité pour la normalisation des données, elle peut être utilisée en ligne de commande via la commande svm-scale. Une telle fonctionnalité peut être intégrée dans l'application dans le cadre d'une amélioration future.

V.3.2 Sélection des paramètres :

La sélection des paramètres SVM est extrêmement importante, dans la mesure où elle influence de façon très significative le processus de prédiction. Les paramètres à choisir ne sont pas identiques à tous les problèmes et ne sont pas forcément connus à l'avance pour un jeu de donné précis, ce qui impose de faire une sélection de paramètres. En d'autres termes il faut rechercher les meilleurs paramètres pour le kernel utilisé de façon à ce que le prédicteur prédise avec une plus grande justesse les valeurs inconnues (données de test).

La LIBSVM propose un utilitaire pour faire de la sélection de paramètres appelé le « Grid-search », qui se base sur la division des données en plusieurs sous-ensembles (n-fold) de plus petite taille pour trouver les meilleurs paramètres C et Gamma.

Les résultats décrits sous-dessous sont obtenus en utilisant un type de SVM : C-SVC et un type de noyau : RBF avec les paramètres par défaut de SVM. Et pour les paramètres de recuit simulé on a choisi une température initiale égale à et le nombre de générations par palier égale à 10.

V.3.3 Résultats :

Les objectifs des expérimentations effectuées sur les trois jeux de données sont de tester l'effet de la sélection des attributs sur l'amélioration du taux de classification. Puisque l'approche proposée est stochastique, nous l'exécutons 15 fois pour chacune des trois jeux de données décrits précédemment.

Les tableaux (Tableaux V.1, V.2 et V.3) montrent l'amélioration des taux de classification pour les trois jeux de données pour chacun des algorithmes SVM sans la sélection des attributs et avec la sélection.

N° Exécution	Breast_Cancer (10 variables)		
	Avec sélection		Taux-Class Sans sélection
	Taux-Class	Nbre-Var-Selc	
1	97,51	8	97,07
2	97,51	7	96,93
3	97,36	8	97,22
4	97,51	7	96,93
5	97,36	6	96,93
6	97,51	7	96,93
7	97,51	9	97,22
8	97,51	7	97,07
9	97,51	8	97,07
10	97,22	7	96,93
11	97,51	8	97,22
12	97,36	7	97,22
13	97,51	7	97,07
14	97,66	7	96,93
15	97,66	7	96,93

Taux -Class (Sans sélection)		
Min	Max	Moy
97,22	97,66	97,48

Taux -Class (Avec sélection)		
Min	Max	Moy
96,93	97,22	97,04

Table V.1 : Taux de classification pour les données Breast_cancer.

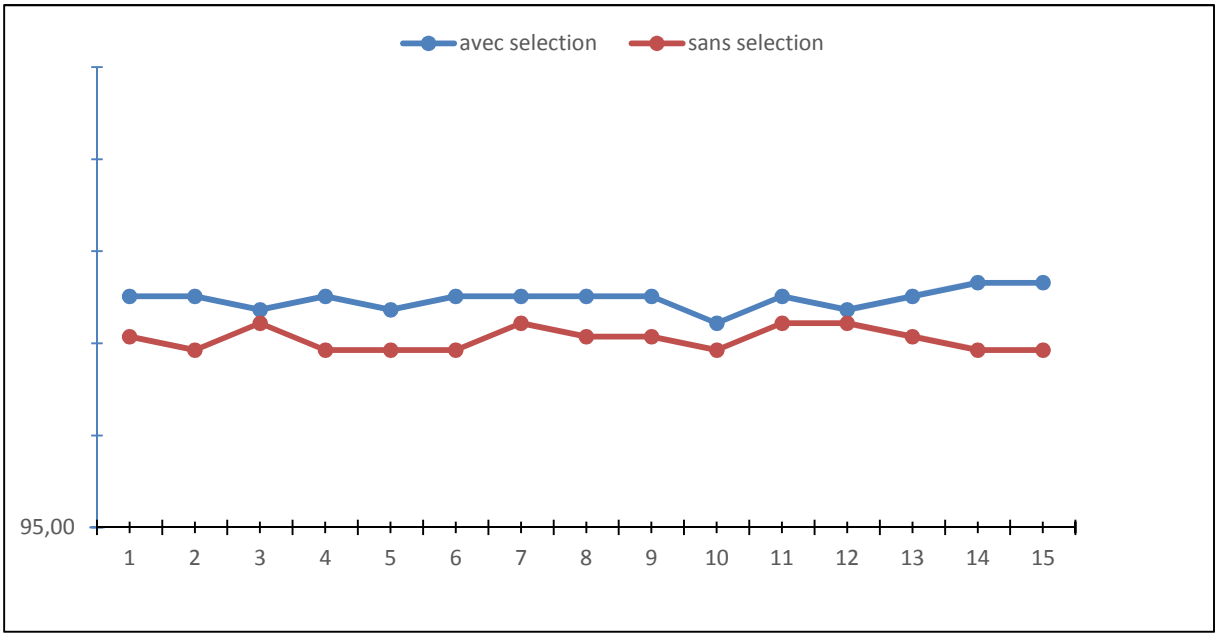


Figure V.3 : Taux de classification pour les données Breast_cancer.

N° Execution	Heart (13 variables)		
	Avec selection		Taux-Class Sans selection
	Taux-Class	Nbre-Var-Selc	
1	85,93	7	82,22
2	84,81	7	81,85
3	86,30	8	82,22
4	84,44	11	82,22
5	85,93	11	83,70
6	86,30	7	82,59
7	85,93	8	82,96
8	85,56	11	82,22
9	84,81	5	81,85
10	85,93	10	81,85
11	84,44	9	81,85
12	86,30	10	82,22
13	86,30	9	81,48
14	86,30	8	82,96
15	85,93	9	82,59

Taux -Class (Sans selection)		
Min	Max	Moy
84,44	86,30	85,68

Taux -Class (Avec selection)		
Min	Max	Moy
81,48	83,70	82,32

Table V.2 : Taux de classification pour les données Heart.

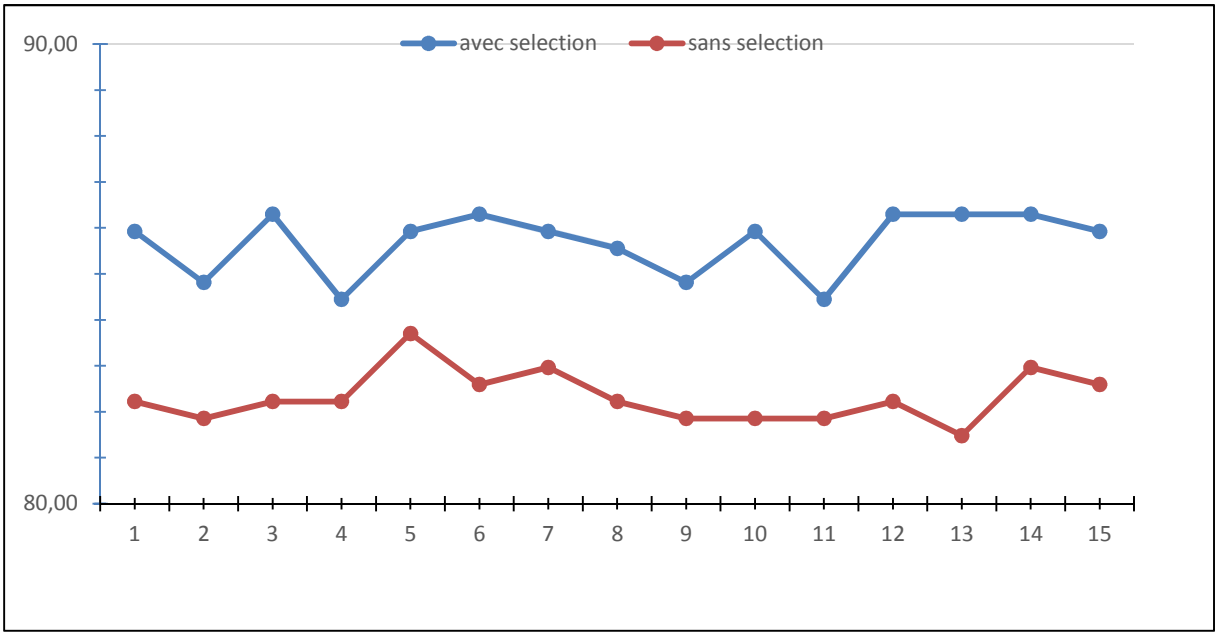


Figure V.4 : Taux de classification pour les données Heart

N° Execution	Diabetes (8 variables)		
	Avec selection		Taux-Class Sans selection
	Taux-Class	Nbre-Var-Selc	
1	78,39	5	76,82
2	78,39	4	77,08
3	78,52	5	76,95
4	77,86	5	77,21
5	78,13	6	77,21
6	78,13	5	77,73
7	78,78	4	77,47
8	78,65	5	77,08
9	78,13	6	77,21
10	78,13	5	77,21
11	77,73	6	77,73
12	77,99	6	76,82
13	78,39	6	76,69
14	78,52	4	77,47
15	78,26	6	77,34

Taux -Class (Sans selection)		
Min	Max	Moy
77,73	78,78	78,26

Taux -Class (Avec selection)		
Min	Max	Moy
76,69	77,73	77,20

Table V.3 : Taux de classification pour les données Diabètes

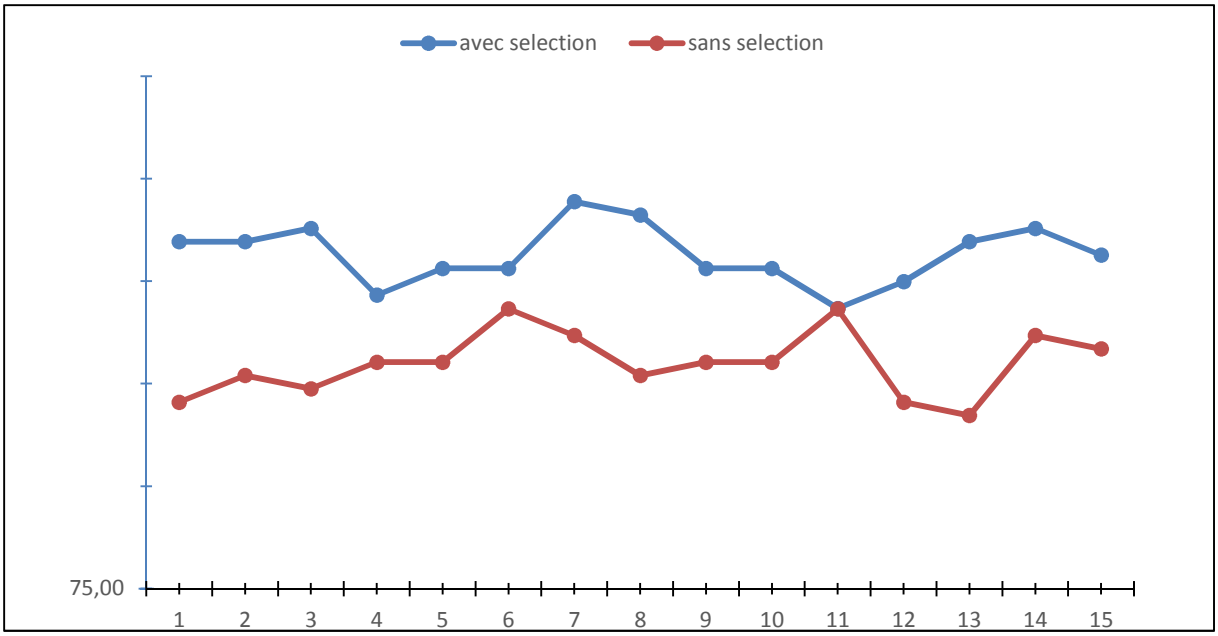


Figure V.5 :Taux de classification pour les données Diabète

L'analyse des tableaux (Tableau V.1, Tableau V.2 et Tableau V.3) et les figures associer (Figure V.3, V.4 et V.5) montre que la sélection des gènes améliore le taux de classification. Cela implique que la sélection de gènes permet effectivement de réduire les données inutiles et le bruit pour améliorer le taux de classification.

5.4 Conclusion :

Nous avons présenté dans ce chapitre l'application réalisée, et nous avons évalué les performances de l'approche proposée sur trois jeux de données du domaine d'oncologie. Les résultats obtenus montrent que la sélection de gènes améliore le taux de classification. Cela implique que la sélection de gènes permet effectivement de réduire les données inutiles et le bruit pour améliorer le taux de classification.

Conclusion générale

La sélection des variables est un domaine de recherche qui donne lieu à de nombreuses études et à de nouvelles approches, Le travail réalisé dans ce mémoire apporte une contribution concernant la sélection de variables pour des problèmes de classification supervisée.

Tout d'abord, nous avons développé une approche de type intégrée pour faire face au problème de la sélection des gènes pour la classification des données des bio-puces.

Cette approche utilise un algorithme de recherche. Pour cela, un algorithme de recherche a été implémenté c'est la recuit simulé. De plus cet algorithme utilise un classifieur SVM multi-classes pour évaluer la qualité d'un sous-ensemble de gènes sélectionnés et pour éliminer les gènes redondants

Enfin, cette étude montre que la méthode proposée est une méthode simple, cependant puissante pour la sélection de gènes et améliore bien le taux de la classification de données de puces à ADN. Son efficacité dépend fortement de comment les informations sémantiques du problème d'apprentissage sont intégrées dans ses opérateurs fondamentaux comme la fonction d'évaluation et le voisinage.

Perspectives de recherche :

Le travail réalisé au cours de ce mémoire nous ont permis de conclure que le problème de la sélection de variables est très prometteur, surtout dans le domaine de sélection de gènes pour les puces à ADN. De ce fait, nos perspectives de recherche sur la sélection de variables porteront, essentiellement, sur la proposition des techniques complémentaires permettant d'améliorer nos résultats sur la sélection de gènes pour la classification de données de bio-puces concernant les problèmes de reconnaissance de cancers le diagnostic en oncologie.

Comme perspective aussi, de proposer une approche intégrée avec la méthode de sélection de gènes qui cherche les meilleurs paramètres C et Gamma du classifieur SVM multi-classes.

Bibliographie

[1] Fiche d'information du CCSM (CENTRE CANADIEN SCIENCE ET MEDIAS), Gènes, ADN et chromosomes.

[2]http://biochimej.univangers.fr/Page2/COURS/9ModulGenFoncVeg/5MethEtudGenFonc/3PucesADN/1PucesADN.htm Date de consultation : 26/04/2015

[3]http://acces.enslyon.fr/acces/ressources/sante/dynamique/dynacell%20plone/cancer/html/pointpuces.htm/ Date de consultation : 27/04/2015

[4] Nancie Reymond, Bioinformatique des puces à ADN et application à l'analyse du transcriptome de Buchnera aphidicola.

[5] José Crispin Hernandez Hernandez. Hybrid metaheuristics algorithms for gene selection and classication of microarray data. Computer Science. Universite d'Angers, 2008. French.

[6] Ali El Akadi, contribution à la sélection de variables pertinentes en classification supervisée : application à la sélection des gènes pour les puces à ADN et des caractéristiques faciales. Université MOHAMMED V, Rabat.

[7] Sabra El Ferchich . Sélection et extraction d'attributs pour les problèmes de classification. Université des sciences technologies Lille 1

[8] Mohamadally Hasan, Fomani Boris .SVM : Machines à Vecteurs de Support ou Séparateurs à Vastes Marges,.

[9]Hamou Reda Mohamed , Ahmed Lehireche . La classification non supervisée (clustering) de documents textuels par les automates cellulaires. University Djillali Liabes of Sidi Bel Abbes

[10] Mouhamadou Lamine SAMB, Fodé CAMARA, Samba NDIAYE, Yahya SLIMANI et Mohamed Amir ESSEGHIR. Approche de sélection d'attributs pour la classification basée sur l'algorithme RFE-SVM. Université Cheikh Anta Diop de Dakar SENEGAL

[11] BENDAHMANE Amine. Le recuit simulé. Université des Sciences et de la Technologie d'Oran - Mohamed Boudiaf

[12] Edmundo Bonilla Huerta. Logique floue et algorithmes génétiques pour le pré-traitement de données de biopuces et la sélection de gènes. Computer Science. Université d'Angers, 2008.

[13] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and cluste-ring. IEEE Trans. on Knowledge and Data Engineering, 17(4):491–502, 2005.

[14] R. Kohavi and G.H. John. Wrappers for feature subset selection. Artif. Intell., 97(12):273–324,1997.

[15] D. Koller and M. Sahami. Toward optimal feature selection. 13th International conference on machines learning, pages 1–15, 1996.

[16] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research, 5:1205–1224, 2004.

[17] Bennani ,y.(2001).systèmes d'apprentissage connexionnistes : sélection de variables. Revue d'intelligence artificielle .

[18] http://www.transcriptome.ens.fr/sgdb/presentation/principle.php Date de consultation : 02/05/2015

[19] Baptiste Autin. Les métaheuristiques en optimisation combinatoire, Conservatoire National Des Arts et Metiers Paris .

[20] BENDAHOUA Sarah, le recuit simulé, Université des Sciences et de la Technologie d'Oran - Mohamed Boudiaf